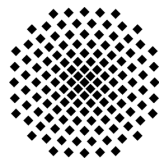


Bachelorarbeit

# Modellierung eines 2D diskretisierten Wärmeübertragers zur Kühlung von Leistungselektronik

S A S K I A   D I E T M A I E R



**Universität Stuttgart**

Universität Stuttgart

Institut für Thermodynamik der Luft- und Raumfahrt (ITLR)

Direktor: Prof. Dr.-Ing. habil. Bernhard Weigand

# Zusammenfassung

Diese Arbeit umfasst das Modell eines zweidimensional diskretisierten Wärmeübertragers sowie einer Metallbibliothek in Modelica. Die Form des Wärmeübertragers entspricht einer „Tubed Cold Plate“ welche in den Flüssigkühlkreislauf zur Kühlung von Leistungselektronik innerhalb des „More Electric Aircraft“ verwendet wird.

Die Diskretisierung des Wärmeübertragers erfolgt nach dem Verfahren der Finiten- Volumen. Dabei wird zwischen Flüssig- und Festkörperelementen differenziert, wobei für jedes Element Erhaltungssätze definiert sind. Um dem Modell hinsichtlich der Geometrie und Anzahl der verlaufenden Kühlkanäle Flexibilität zu gewährleisten, ist die Verschaltung der Elemente parametrisiert aufgebaut. Dabei werden die einzelnen Zellen in Arrays definiert, wobei die Verschaltung mittels *if*- Bedingungen und *for*- Schleifen stattfindet. Für die Wärmeübergänge sind geeignete Korrelationen für die Konvektion am Festkörperelement mit der Umgebung an den verschiedenen Flächen der Elemente vorhanden. Diese können durch das Modell einer Konvektion, welches die Stoffwerte der Umgebung enthält, dem Wärmeübertrager übergeben werden. Zusätzlich sind für die laminare und turbulente Rohrströmung Korrelationen für Konvektion und Druckverlust implementiert.

In der Metallbibliothek sind Stoffwerte von verschiedenen Metallen tabellarisch aufgelistet, die bei Aufruf interpoliert werden. Diese werden sowohl für das Festkörperelement als auch für das Rohr des Flüssigelements benötigt.

Die Validierung des Wärmeübertragermodells erfolgt zum einen an einem Modell, welches sich hinsichtlich der Umgebung adiabat verhält, aber dennoch mit der Leistungselektronik verbunden ist, und am realen Modell, welches mit der Elektronik und der Umgebung Wärme austauscht, und wird hinsichtlich der Rechenzeit und der Funktionalität des Wärmeübergangs validiert.

Die Leistungselektronik wird dabei einmal als eine heiße Temperatur modelliert, die mit der Oberseite der Elemente verbunden ist, und beim zweiten Mal als vorgeschriebener Wärmestrom, der den Zellen übergeben wird.

Für die „Tubed Cold Plate“ werden Visualisierungen erstellt, um den Wärmeübergang und somit die Temperaturänderung der Elemente bei verschiedener Diskretisierung und Geometrie visuell darzustellen.

Diese beiden neuen Modelica Bibliotheken sind in die DLR- interne Bibliothekensammlung integriert und dienen zur Vorauslegung von Wärmeübertragern.

# Inhaltsverzeichnis

Abbildungsverzeichnis	V
Tabellenverzeichnis	VII
Symbolverzeichnis	VIII
<b>1 Einleitung</b>	<b>1</b>
1.1 Stand der Technik . . . . .	1
1.2 Motivation und Zielsetzung der Arbeit . . . . .	1
<b>2 Software</b>	<b>3</b>
2.1 Modellierungssprache Modelica . . . . .	3
2.2 Simulationsumgebung Dymola . . . . .	3
2.2.1 Standardbibliothek . . . . .	5
2.2.2 AirSystemsMedia Bibliothek . . . . .	6
2.2.3 FluidDynamics Bibliothek . . . . .	6
2.2.4 Visualization Bibliothek . . . . .	6
<b>3 Geometrie des Wärmeübertragers</b>	<b>7</b>
3.1 Base Record . . . . .	7
3.2 Cold Plate Modell . . . . .	10
<b>4 Transportphänomene</b>	<b>11</b>
4.1 Konvektion am Festkörperelement . . . . .	11
4.1.1 Modell für Konvektion . . . . .	11
4.1.2 Konstanter Wärmeübergang . . . . .	12
4.1.3 Freie Konvektion . . . . .	13
4.1.3.1 Vertikale Platte . . . . .	13
4.1.3.2 Horizontale Platte . . . . .	13
4.1.4 Erzwungene Konvektion . . . . .	14
4.1.5 Freie Konvektion am horizontalen Zylinder . . . . .	15
4.2 Konvektion der Rohrströmung . . . . .	15
4.2.1 Modell für Konvektion . . . . .	15
4.2.2 Konstanter Wärmeübergang . . . . .	16
4.2.3 Erzwungene Konvektion . . . . .	16
4.3 Druckverluste im Rohr . . . . .	17
4.3.1 Modell für Druckverluste . . . . .	18

4.3.2	Ohne Druckverluste . . . . .	18
4.3.3	Konstante Druckverluste . . . . .	18
4.3.4	Druckverluste durch Rohrreibung . . . . .	19
4.3.5	Druckverluste für die Änderung der Strömungsrichtung . . . . .	20
<b>5</b>	<b>Zellen</b>	<b>21</b>
5.1	Festkörperelement . . . . .	21
5.1.1	Modell für Festkörperzelle . . . . .	21
5.1.2	Festkörperzelle: SolidCellEdge . . . . .	22
5.1.3	Festkörperzelle: SolidCellCenter . . . . .	23
5.2	Flüssigelement . . . . .	24
5.2.1	Modell für Flüssigzelle . . . . .	24
5.2.2	Flüssigzelle . . . . .	25
5.2.3	Flüssigzelle der Rohrkrümmung . . . . .	27
<b>6</b>	<b>Metall Materialbibliothek</b>	<b>28</b>
<b>7</b>	<b>Umgebung</b>	<b>31</b>
7.1	Base Record . . . . .	31
7.2	Stoffwerte der Umgebung . . . . .	32
7.3	Konvektion . . . . .	33
7.4	Konvektion Umgebung . . . . .	34
<b>8</b>	<b>Diskretisierung des Wärmeübertragers</b>	<b>35</b>
<b>9</b>	<b>Diskussion und Bewertung der Ergebnisse</b>	<b>41</b>
9.1	Validierung der Flüssigzelle . . . . .	41
9.2	Untersuchung der verschiedenen Lösungsalgorithmen . . . . .	42
9.3	Validierung des adiabaten Wärmeübertragers . . . . .	45
9.4	Überprüfung der Funktionalität der Diskretisierung . . . . .	46
9.5	Modell mit vorgeschriebenen Wärmestrom . . . . .	49
<b>10</b>	<b>Visualisierung</b>	<b>51</b>
<b>11</b>	<b>Zusammenfassung und Ausblick</b>	<b>52</b>
	<b>Literaturverzeichnis</b>	<b>53</b>
	<b>Anhang</b>	<b>55</b>
.1	Metallbibliothek . . . . .	55
.1.1	Aluminium und Aluminiumlegierungen . . . . .	55
.1.2	Kupfer und Kupferlegierungen . . . . .	55
.1.3	Eisen und Eisenlegierungen . . . . .	56
.1.4	Silber . . . . .	57
.1.5	Stahl und Stahllegierungen . . . . .	57
.1.6	Edelstahllegierungen . . . . .	58

## INHALTSVERZEICHNIS

.1.7	Silizium . . . . .	59
.2	Farbschema Jet . . . . .	59

# Abbildungsverzeichnis

1.1	„Tubed Cold Plate“ . . . . .	2
2.1	Code: Pendel Modell . . . . .	4
2.2	Code: Aufruf eines <code>replaceable</code> package . . . . .	5
3.1	Geometriedaten des Wärmeübertragers . . . . .	7
4.1	Struktogramm Konvektion am Festkörperelement . . . . .	12
4.2	Struktogramm Konvektion der Rohrströmung . . . . .	16
4.3	Struktogramm Druckverluste im Rohr . . . . .	18
5.1	Festkörperelement . . . . .	22
5.2	Flüssigelement . . . . .	25
5.3	Flüssigelement der Rohrkrümmung . . . . .	27
7.1	Struktogramm Umgebung . . . . .	31
8.1	Diskretisierung des Wärmeübertragers . . . . .	35
8.2	Code: Verbindung <code>LiquidCell</code> mit <code>liquidIn-Outlet</code> . . . . .	36
8.3	Code: Verbindung der Flüssigzellen . . . . .	37
8.4	Code: Verbindung <code>LiquidCell</code> mit <code>LiquidCellChangeOfStream</code> . . . . .	37
8.5	Code: Verbindung <code>LiquidCell</code> mit Elektronik . . . . .	38
8.6	Code: Verbindung <code>SolidCellEdge1</code> . . . . .	38
8.7	Code: Verbindung <code>SolidCellCenter</code> . . . . .	40
9.1	Validierung der Flüssigzelle: Exponentieller bzw. linearer Temperaturanstieg bei konstanter Temperaturbelastung . . . . .	42
9.2	Validierung der Flüssigzelle: Quadratischer Temperaturanstieg bei linearer Temperaturbelastung . . . . .	42
9.3	Testmodell . . . . .	43
9.4	Rechenzeit von unterschiedlichen Lösungsverfahren . . . . .	44
9.5	Überprüfung der Diskretisierung: Temperaturänderung $\Delta T$ des Fluides über $n_{Solid}$ . . . . .	46
9.6	Überprüfung der Diskretisierung: Wärmestrom $\dot{Q}$ des Fluides über $n_{Solid}$ . . . . .	47
9.7	Überprüfung der Anzahl der Passes mit $n_{Solid} = 5$ . . . . .	47
9.8	Genauere Betrachtung der Diskretisierung bei $n_{passes} = 2$ . . . . .	47
9.9	Zelltemperaturen des Zentrums zwischen zwei Kühlkanälen . . . . .	48
9.10	Rechenzeit CPU des Fluides über $n_{Solid}$ . . . . .	49
9.11	Rechenzeit CPU des Fluides über $n_{passes}$ bei $n_{Solid} = 5$ . . . . .	49

9.12	Temperaturänderung $\Delta T$ bei vorgeschriebenen Wärmestrom $\dot{Q}$ der Elektronik des Fluides über $n_{Solid}$ . . . . .	50
9.13	Temperaturänderung $\Delta T$ bei vorgeschriebenen Wärmestrom $\dot{Q}$ der Elektronik des Fluides über $n_{passes}$ bei $n_{Solid} = 1$ . . . . .	50
10.1	Visualisierung $n_{Solid} = 3$ und $n_{passes} = 4$ . . . . .	51
10.2	Visualisierung $n_{Solid} = 12$ und $n_{passes} = 4$ . . . . .	51
.1	Farbschema Jet . . . . .	59

# Tabellenverzeichnis

4.1	Parameter $C$ und $n$ für die freie Konvektion am horizontalen Zylinder . .	15
6.1	Stoffwerte von reinem Aluminium . . . . .	29
6.2	Stoffwerte von reinem Kupfer . . . . .	29
6.3	Stoffwerte von reinem Aluminium und Kupfer mit $c_p = 0$ und $\lambda = konst$	30
7.1	Stoffwerte der Umgebung . . . . .	32
7.2	Stoffwerte von $c_{pDry}$ und $c_{pVapour}$ . . . . .	33
9.1	Nachweis der gleichen Wärmeströme bei $n_{Solid} = 3$ und $n_{passes} = 3$ . . . .	45
.1	Stoffwerte von Aluminium 2024T6 . . . . .	55
.2	Stoffwerte von Aluminium 195 Guss . . . . .	55
.3	Stoffwerte von kommerzieller Bronze . . . . .	55
.4	Stoffwerte von Phosphor Bronze . . . . .	56
.5	Stoffwerte von Messing . . . . .	56
.6	Stoffwerte von Konstantan . . . . .	56
.7	Stoffwerte von reinem Eisen . . . . .	56
.8	Stoffwerte von Armco . . . . .	56
.9	Stoffwerte von Silber . . . . .	57
.10	Stoffwerte von gewöhnlichen Stahl . . . . .	57
.11	Stoffwerte von AISI 1010 . . . . .	57
.12	Stoffwerte von Kohlenstoff Silizium . . . . .	57
.13	Stoffwerte von Kohlenstoff Mangan Silizium . . . . .	58
.14	Stoffwerte von AISI 302 . . . . .	58
.15	Stoffwerte von AISI 304 . . . . .	58
.16	Stoffwerte von AISI 316 . . . . .	58
.17	Stoffwerte von AISI 347 . . . . .	58
.18	Stoffwerte von Silizium . . . . .	59



# Symbolverzeichnis

## Lateinische Symbole

$A$	$m^2$	Fläche einer Zellseite
$C$	$J/K$	Wärmekapazität
$c_p$	$J/(kg\ K)$	spezifische Wärmekapazität bei konstantem Druck
$CPU$		Central Processing Unit
$dh$	$J/kg$	Enthalpieänderung zwischen zwei FluidPorts
$d$	$m$	Durchmesser der Flüssigzelle
$g$	$m/s^2$	Erdbeschleunigung
$Gr$		Grashofzahl
$h$	$J/kg$	spezifische Enthalpie
$HTML$		Hypertext Markup Language
$k$	$W/(m^2\ K)$	Wärmedurchgangszahl
$L$	$m$	charakteristische Länge
$l$	$m$	Länge einer Flüssigzelle
$\dot{m}$	$kg/s$	Massenfluss
$n$		Diskretisierung, Anzahl der Elemente
$Nu$		Nusseltzahl
$p$	$Pa$	Druck
$PlateLength$	$m$	Länge der Platte
$PlateWidth$	$m$	Breite der Platte
$Pr$		Prandtlzahl
$\dot{Q}$	$W/s$	Wärmestrom
$R$	$^\circ$	Innenradius der LiquidCellChangeOfStream
$Ra$		Rayleighzahl
$Re$		Reynoldszahl
$T$	$K$	Temperatur
$U$	$m^3$	Umfang
$V$	$m^3$	Volumen
$v$	$m/s$	Geschwindigkeit
$w$	$m$	Dicke/Höhe
$width$	$m$	Breite eines Plattenabschnitts
$Xi$	$kg/kg$	Massenanteil
$x$	$m$	Breite einer Festkörperzelle
$y$	$m$	Länge einer Festkörperzelle

$z$	$m$	Tiefe einer Festkörperzelle
-----	-----	-----------------------------

## Griechische Symbole

$\alpha$	$W/(m^2 K)$	Wärmeübergangszahl
$\beta$	$1/K$	isochorer Spannungskoeffizient
$\gamma$		Hilfsfunktion
$\Delta$		Änderung
$\delta$	$^\circ$	Öffnungswinkel der LiquidCellChangeOfStream
$\eta$	$kg/(m s)$	dynamische Viskosität
$\lambda$	$W/(m K)$	Wärmeleitfähigkeit
$\mu$	$^\circ$	Öffnungswinkel der LiquidCell
$\nu$	$m^2/s$	kinematische Viskosität
$\xi$		Hilfsfunktion
$\varrho$	$kg/m^3$	Dichte
$\varphi$		Hilfsfunktion in Abhängigkeit der Prandtlzahl
$\chi$	$kg_{Water}/kg_{Dry}$	Dampfgehalt

## Indizes

<i>Ambient</i>	Umgebung
<i>Cell</i>	Festkörperelement
<i>center</i>	gesamter Zentrumsbereich
<i>Channel</i>	Kühlmedium
<i>channel</i>	äußere Seite des Kühlrohrs
<i>Dry</i>	trockene Luft
<i>edge</i>	ein Randbereich
<i>erzw</i>	erzwungene Konvektion
<i>friction1</i>	Reibung in LC
<i>friction2</i>	Reibung in LCCOS
<i>grease</i>	Wärmeleitpaste
<i>in</i>	eingehender Port
<i>LC</i>	LiquidCell
<i>LCCOS</i>	LiquidCellChangeOfStream
<i>Liquid</i>	innere Seite des Kühlrohrs
<i>laminar</i>	laminare Strömung
<i>L, 2300</i>	laminare Strömung bei $Re = 2300$
<i>T, 10000</i>	turbulente Strömung bei $Re = 10000$
<i>m, q, 1</i>	Asymptote 1, mittlere Nusseltzahl für kleine Werte von $Re Pr d/L$
<i>m, q, 2</i>	Asymptote 2, mittlere Nusseltzahl für kleine Werte von $Re Pr d/L$
<i>out</i>	ausgehender Port

# SYMBOLVERZEICHNIS

<i>outflow</i>	aus dem Port herausfließend
<i>passes</i>	Durchläufe des Kühlkanals
<i>PE</i>	Elektronik (Power Electronics)
<i>phX</i>	Zustand bei vorgegebenen Druck, Enthalpie, Dampfgehalt
<i>PlateLength</i>	Länge der Platte
<i>SCC</i>	<b>SolidCellCenter</b>
<i>SCE</i>	<b>SolidCellEdge</b>
<i>start</i>	Startwert
<i>Solid</i>	Rand- und/oder Zentrumsbereich
<i>SolidCenter</i>	Zentrumsbereich der Platte
<i>SolidEdge</i>	Randbereich der Platte
<i>turbulent</i>	turbulente Strömung
<i>Vapour</i>	Wasserdampf
<i>Wall</i>	Rohrwand
<i>xy</i>	Fläche xy
<i>xz</i>	Fläche xz
<i>x1</i>	HeatPort x1
<i>x2</i>	HeatPort x2
<i>yz</i>	Fläche yz
<i>y1</i>	HeatPort y1
<i>y2</i>	HeatPort y2
<i>z</i>	HeatPort z
<i>z1</i>	HeatPort z1
<i>z2</i>	HeatPort z2
$\infty$	Fluid in der Mitte des Kanals

# 1 Einleitung

Elektronische Geräte erzeugen stets Wärme. Um einer Überhitzung der Komponenten vorzubeugen, muss die Wärme abgetragen werden. Die gängigsten Methoden sind Luft- und Flüssigkühlung. Meist ist eine Kühlung durch Luft bei Leistungselektronik in Flugzeugen nicht mehr ausreichend, sondern die produzierte Wärme muss durch Wärmeübertrager mit Flüssigkühlung abgetragen werden.

## 1.1 Stand der Technik

Wärmeübertrager sind in vielen verschiedenen Konfigurationen kommerziell erhältlich [1]. So gibt es zum Beispiel „Fin Cold Plates“, „Flat Tube Cold Plates“ oder die in dieser Arbeit modellierte „Tubed Cold Plate“.

Wesentlicher Unterschied dieser Arten ist die Geometrie des Kühlkanals. Bei der „Tubed“ Konfiguration fließt das Kühlmedium durch ein Rohr, wohingegen es bei der „Fin“ Konfiguration durch kleine Lamellen strömt [2].

Bereits 1997 wurden im Zuge der Demonstration der Modellierungssprache Modelica erste Modelle von Wärmeübertragern in Modelica implementiert [3]. Bei diesen Modellen handelt es sich um einfache Gleich- oder Gegenstrom Wärmeübertrager. Zu diesem Zeitpunkt gab es für Modelica allerdings noch keine Simulationsumgebung die mit den variablen Array Einträgen einer Komponente umgehen konnte [4].

Im Jahr 2008 wurde von der Technischen Universität Hamburg- Harburg eine Power Electronic Cooling Library erstellt [5]. Dabei wurden für die Fest- und Flüssigelemente Komponenten aus der Modelica Standardbibliothek verwendet.

## 1.2 Motivation und Zielsetzung der Arbeit

Ziel dieser Arbeit ist es, einen möglichst parametrisiert aufgebauten Wärmeübertrager in Form einer sogenannten „Tubed Cold Plate“ mit Modelica zu erstellen um ein Höchstmaß der Flexibilität hinsichtlich der Geometrie zu erreichen. Dabei soll eine DLR- interne Wärmeübertrager- Bibliothek erstellt werden, welche der Vorauslegung von Wärmeübertragern dienen soll.

Für diesen Zweck ist es notwendig die „Cold Plate“ zu diskretisieren, so dass sie aus Festkörperelementen, welche die Platte darstellen, und aus Flüssigelementen, die den

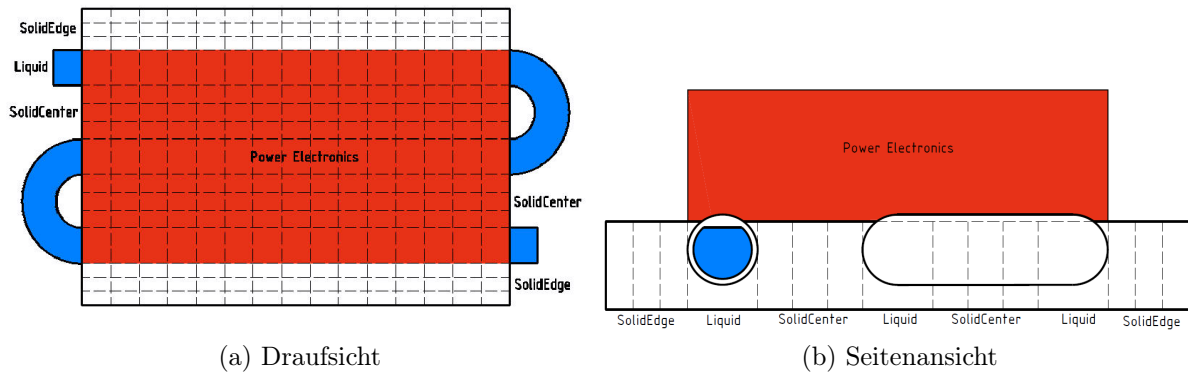


Abbildung 1.1: „Tubed Cold Plate“

Kühlkanälen mit Kühlmittel entsprechen, aufgebaut ist. Es entsteht eine Geometrie, welche in Zellen eingeteilt ist, bei der für jede Zelle eine Temperatur ermittelt wird. Das Grundmodell ist in Abbildung 1.1 zu sehen.

Bei den Festkörperelementen ist es notwendig die Wärmeleitung durch eine ebene Wand zu implementieren. Außerdem benötigen die Randelemente geeignete Korrelationen für die freie und erzwungene Konvektion an deren Außenseite.

Für die Flüssigelemente wird der Wärmeübergang durch die Rohrwand und das Kühlmittel benötigt. Durch Reibung im Rohr geht im Kühlmittelstrom Druck verloren, wodurch auch der Druckverlust im Rohr implementiert wird.

Wärmestrahlung soll im Modell dieser Arbeit vernachlässigt werden.

Die Diskretisierung soll so erfolgen, dass durch Vorgabe von wenigen Parametern, wie zum Beispiel Plattengröße und Anzahl der Zellen, die Zellen sich so miteinander verbinden, dass eine realistische Geometrie eines Wärmeübertragers entsteht.

Des Weiteren wird eine Bibliothek für Metalle erstellt, aus denen die Stoffwerte der Materialien des Wärmeübertragers interpoliert werden können. Für die Konvektion mit der Umgebung und deren Stoffwerte wird ein weiterer Block implementiert.

Durch die objektorientierte Programmierung mit Modelica ist das gesamte Modell der „Cold Plate“ strukturiert in einzelne Elemente aufgebaut, welche hierarchisch voneinander erben und Parameter und Variablen übergeben. Das vollständige Modell des Wärmeübertragers setzt sich folglich aus vielen Paketen und Modellen zusammen. So besteht das Modell aus den folgenden Paketen:

- Geometrie des Wärmeübertragers
- Transportphänomene: Funktionen für Konvektion und Druckverlust
- Zellen
- Umgebung
- Metallbibliothek
- Diskretisierter Wärmeübertrager

Abschließend soll das „Cold Plate“ Modell hinsichtlich der Orts- und Zeitdiskretisierung überprüft werden und die Temperaturverteilung der einzelnen Zellen visualisiert werden.

## 2 Software

Dieses Kapitel soll einen kurzen Überblick über das objektorientierte Programmieren mittels der Modellierungssprache Modelica und der Simulationsumgebung Dymola geben. Außerdem werden die vorhandenen Bibliotheken, welche in dieser Arbeit verwendet werden, beschrieben.

### 2.1 Modellierungssprache Modelica

Modelica ist eine objektorientierte und gleichungsbasierte Programmiersprache die 1996 durch die Modelica Association entworfen wurde. Die Modelica Association ist eine gemeinnützige Organisation bei der verschiedene internationale Teams aus Europa, Kanada, USA und Asien zusammenarbeiten [6].

Modelica eignet sich besonders zur Modellbildung komplexer Systeme und zur multiphysikalischen Modellierung von verschiedenen ingenieurtechnischen Bereichen. So können zum Beispiel mechanische und elektrische Interaktionen realitätsnah in einem Modell simuliert werden. Es gibt Standardbibliotheken, in denen Modelle und Funktionen beschrieben sind, die durch einfaches „Ziehen und Ablegen“ verwendet werden können. Diese Bibliotheken werden stets weiterentwickelt und Neue geschaffen, so dass sich die Modellierungssprache Modelica in stetiger Entwicklung befindet.

Physikalische Modelle und Komponenten werden durch Zustands- und Erhaltungsgleichungen beschrieben. Diese können in einer graphischen Benutzeroberfläche mittels Konnektoren verbunden werden und voneinander hierarchisch erben.

Wird die Modellierungssprache Modelica verwendet, wird der Code mit Hilfe eines Compilers übersetzt und durch einen Solver numerisch berechnet. Solche numerischen Lösungsverfahren stellen Simulationsprogramme bereit. Diese gibt es sowohl kostenlos, zum Beispiel OpenModelica, als auch kommerziell, wie Wolfram SystemModeler und Dymola [7].

### 2.2 Simulationsumgebung Dymola

Das Dynamic Modeling Laboratory, kurz Dymola, von Dassault Systems ist ein kommerzielles Simulationsprogramm, welches es bereits schon vor der Entstehung von Modelica gab [8]. Für den Gebrauch von Dymola benötigt man einen C- Compiler, da Dymola

den Modelica Code in einen C- Quellcode übersetzt und dadurch das System verkleinert und effizienter macht.

Für die Modellierung und Simulation stehen in Dymola ein Modellierungsmodus und ein Simulationsmodus zur Verfügung. Der Modellierungsmodus bietet vier verschiedene Möglichkeiten zur Bearbeitung des Modells.

Die *Icon* Darstellung, welche der einfachen Erstellung eines grafischen Symbols einer Komponente dient. Die *Diagramm* Darstellung, in welcher die verschiedenen Komponenten eines Modells durch Icons dargestellt werden, und durch Konnektoren grafisch verbunden werden können. Die *Dokumentations* Darstellung, in welcher Informationen der Komponente oder des Modells in HTML beschrieben sind. Und die *Text* Darstellung, in der gleichungsbasiert programmiert werden kann. Die *Text* Darstellung wird bei der Erstellung von neuen Komponenten verwendet, da nur in dieser Ebene ein System mathematisch und physikalisch beschrieben werden kann.

Vor der Simulation kann durch einen „Check“ Knopf überprüft werden, ob das Modell funktionsfähig ist, zum Beispiel ob es kausal ist, es also genauso viele Gleichungen wie Unbekannte enthält [7].

Zur Simulation stehen in Dymola unterschiedliche numerische Lösungsverfahren, wie zum Beispiel *Dassl*, *Euler* oder *Runge-Kutta* zur Verfügung. Im Simulationsmodus kann die Simulationsdauer bestimmt werden und nach der Simulation können die einzelnen Variablen über die Zeit in Tabellen oder Graphen angezeigt werden.

Wird ein Modell in der *Text* Darstellung programmiert, muss dies entsprechend Modelica Syntax erfolgen. Hierfür folgendes Beispiel eines einfachen Pendels.

---

```

1  model Pendel
2      constant Modelica.SIunits.Acceleration g=9.81;
3      parameter Modelica.SIunits.Length l=1;
4      Modelica.SIunits.Angle phi(start=Modelica.Constants.pi*0.9);
5      Modelica.SIunits.AngularVelocity omega(start=0);
6  equation
7      der(phi)=omega;
8      der(omega)=-(g/l)*Modelica.Math.sin(phi);
9  end Pendel;
```

---

Abbildung 2.1: Code: Pendel Modell

Das Modell wird mit `model` und dem Namen des Modells begonnen und mit `end` beendet. Zunächst erfolgt zu Beginn des Modells eine Deklaration aller vorkommender Konstanten, Parameter und Variablen. Konstanten und Parameter werden jeweils mit `constant` und `parameter` deklariert. Variablen benötigen eine solche Bezeichnung nicht. Hilfreich ist es, Konstanten, Variablen und Parameter mit der Einheit zu benennen, welche unter `Modelica.SIunits` in der Standardbibliothek zu finden sind.

Schließlich werden die Gleichungen unter `equation` (Zeile 6) eingeführt.

Als Deklaration einer Programmklasse gibt es in Modelica unter anderem die Klassen `model`, `record`, `function` sowie `connector` und `package`.

Ein `record` ist eine bestimmte Form, einer Klasse, in der lediglich Parameter aufgelistet werden und in dem keine Gleichungen stehen. Mehrere Klassen werden zu einem

**package** zusammengefügt um eine geordnete Struktur zu erhalten. Dadurch werden zum Beispiel ähnliche Modelle, wie die verschiedenen Korrelationen für die Konvektion, in einem Paket zusammengefasst. In einer **function** werden Funktionen beschrieben, bei denen Gleichungen statt mit **equation** mit **algorithm** angegeben werden. Außerdem werden hier die Eingaben für die Funktionen als **inputs** beschrieben und das was die Funktion berechnet und ausgibt mit **output**. Durch die Klasse **connector** werden physische Konnektoren beschrieben.

Charakteristisch für die Programmierung und Modellierung mit Dymola ist die hierarchische Vererbung der einzelnen Teilmodelle. Jedes Modell ist in sich konsistent, das heißt es enthält genauso viele Parameter wie Gleichungen. Durch die Vererbung eines Modells mittels des Befehls **extends**(Name des Modells) werden alle Parameter, Variablen und Gleichungen dem neuen Modell übergeben, welches letztlich auch in sich konsistent sein muss. Die Klasse von der geerbt wird, wird als partiales Grundmodell beschrieben [9]. In einem **partial model** werden genauso wie bei **function** alle Parameter und Variablen als **inputs** bezeichnet und es kann auch einen **output** geben. Im **partial model** werden alle Parameter und Variablen deklariert, welche in den Modellen die von diesem Modell erben, benötigt werden.

Wird ein solches Modell einem anderen Modell übergeben, geschieht dies durch folgenden Aufruf, zum Beispiel eines bestimmten Materials:

---

```

1  replaceable package SolidMedium =
2                                MaterialTableBased . Metall . Aluminium . AluminiumPure
3  constrainedby                 MaterialTableBased . TableBased

```

---

Abbildung 2.2: Code: Aufruf eines **replaceable package**

Als *SolidMedium* ist somit *AluminiumPure* eingestellt, welches von dem **model** *TableBased* erbt. Dadurch kann im UserInterface das *SolidMedium* geändert werden, wobei alle Medien zur Auswahl stehen, welche von dem **partial model** *TableBased* erben.

Zur Einbindung eines **records** und **models** gibt es ebenfalls die **replaceable** Funktion.

### 2.2.1 Standardbibliothek

Die Modelica Standardbibliothek ist ein kostenloses Paket, das von der Modelica Association entwickelt wird. Sie beinhaltet 1280 Modelle und Blöcke sowie 910 Funktionen zu verschiedenen ingenieurtechnischen Bereichen [6]. Als Blöcke werden in Modelica Ein- oder Ausgabesignale bezeichnet welche Werte übergeben.

Die Standardbibliothek enthält unter anderem Einheiten, Konstanten wie zum Beispiel die Kreiszahl Pi sowie mathematische Funktionen wie beispielsweise den natürlichen Logarithmus. Andere Bibliotheken bauen meist auf der Standardbibliothek auf und enthalten Elemente aus dieser.

Außerdem beinhaltet sie ein Magnet-, Mechanik-, Fluid-, Wärme- und Medienpaket. Für diese Arbeit sind das Fluid- und Wärmepaket wichtig. So stammen die **HeatPorts**,



welche die Wärmeübergänge zwischen den Elementen darstellen, aus dem Wärmepaket. Der **HeatPort** besitzt eine Temperaturvariable und eine Wärmestromvariable. Ist der **HeatPort** mit einem anderen Port verbunden, so besitzen die beiden Ports dieselbe Temperatur. Ist der Port nicht verbunden, hat er keine Eingänge und das System ist adiabat.

Im Medienpaket gibt es verschiedene Medien für Luft, Gase oder Wasser. Diese beruhen alle auf dem **PartialMedium**, welches alle Deklarationen eines Mediums enthält. Hierauf wird auch die Metallbibliothek aufgebaut.

### 2.2.2 AirSystemsMedia Bibliothek

Die am DLR entwickelte **AirSystemsMedia** Bibliothek enthält verschiedene Medien für Luft und Flüssigkeiten. Für Luft gibt es trockene und feuchte Luft. Als flüssige Medien stehen einem das Kühlmittel Galden HT135, PGW (Propylen Glykol Wasser) und Kerosin (Jet A und Jet A1) zur Auswahl.

Auch diese Medien beruhen auf dem **PartialMedium** aus der Standardbibliothek und erben hierarchisch von diesem.

### 2.2.3 FluidDynamics Bibliothek

In der **FluidDynamics** Bibliothek sind unter anderem Konnektoren für Luft, Kühlmittel und für Flüssigkeiten enthalten. Die in dieser Arbeit verwendeten Konnektoren für den Kühlmittelstrom sowie der In- und Outlet und die Quelle und Senke stammen aus dieser Bibliothek.

Der Konnektor **LiquidPort** beinhaltet eine Auswahl für das Kühlmedium und wird durch Massenfluss, Druck, Enthalpie sowie dem Massenanteil beschrieben.

Auch die Quelle und Senke für den Kühlmittelstrom beinhalten die Auswahl des Mediums. Durch die Quelle und Senke werden die physikalischen Eigenschaften des Kühlmittelstroms bestimmt. Es kann der Druck, die Stärke des Massenfluss sowie die spezifische Enthalpie oder Temperatur vorgegeben werden.

Der In- und Outlet verbindet den **LiquidPort** mit der Quelle beziehungsweise Senke.

Außerdem sind in der **FluidDynamics** Bibliothek Funktionen für die Interpolation von Tabellen, zum Beispiel für die Stoffwerte von Medien, vorhanden.

### 2.2.4 Visualization Bibliothek

Durch das am DLR entwickelte Programm **SimVis**, welches über die Bibliothek **Modelica Visualization** geladen wird, können realitätsnahe Simulationen von zum Beispiel sich bewegenden Oberflächen oder fahrenden Autos und Flugzeugen gemacht werden [10]. Dabei können Kamerapositionen, wie das Objekt betrachtet werden soll, und Umgebungen eingestellt werden, um eine realistische Simulation zu erstellen. Diese Bibliothek wird dazu verwendet um die „Cold Plate“ zu visualisieren.

## 3 Geometrie des Wärmeübertragers

In diesem Kapitel werden die Definitionen und Berechnungen zu den geometrischen Abmessungen der „Cold Plate“ erläutert. Diese werden im Modell der „Tubed Cold Plate“ extern im Geometrie Paket gespeichert.

Das Paket gliedert sich in ein **Base Record**, in dem die Parameter aufgelistet werden und einem „Cold Plate“ **record** in dem den Parametern Werte zugewiesen werden.

### 3.1 Base Record

Die Benennungen der Geometrieparameter sind in Abbildung 3.1 zu sehen. Die Benennungen in rot, sind die Parameter welchen Werte vorgegeben werden müssen, die in schwarz welche berechnet werden.

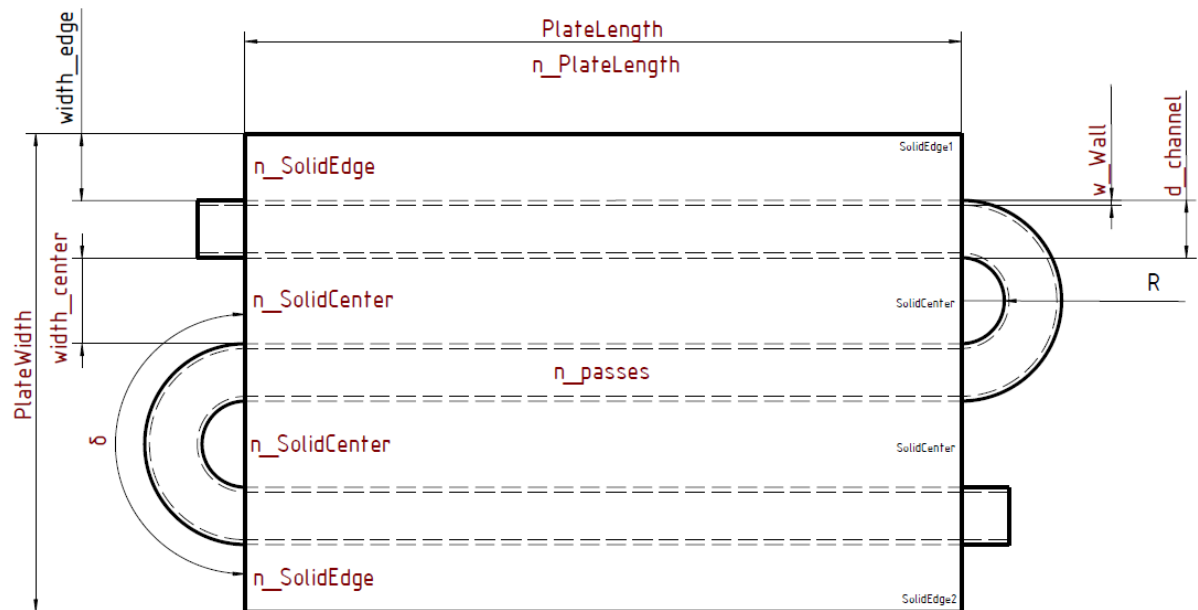


Abbildung 3.1: Geometriedaten des Wärmeübertragers. Die Werte in rot müssen vorgegeben werden, die schwarzen werden berechnet.

Das **record** *BaseRecord* beinhaltet die Parameter für die Dimension des Wärmeübertragers, also Länge (*PlateLength*) und Breite (*PlateWidth*) der Platte, sowie die Anzahl

der diskretisierten Elemente (in die Länge  $n_{PlateLength}$  und die Bereiche  $n_{SolidEdge}$  und  $n_{SolidCenter}$ ). Auch die Größe des **SolidCenters** muss durch  $width_{center}$  vorgegeben werden.

Die Abmessungen des Kühlkanals sind durch  $d_{channel}$ ,  $w_{Wall}$ ,  $\mu$  und der Anzahl der Durchläufe des Kühlkanals  $n_{passes}$  gegeben, siehe Abb. 5.2. Durch Änderung von  $n_{passes}$  kann die Platte eine völlig andere Geometrie erhalten.

Alle Zellen haben über die Länge die gleiche Diskretisierung. In der Breite wird zwischen **SolidCellEdge**, **SolidCellCenter** und **LiquidCells** unterschieden. Flüssigzellen sind in radialer Richtung nur eindimensional und werden bei der Änderung der Strömungsrichtung, am Ende der Platte durch die **LiquidCellChangeOfStream** verbunden.

Die **SolidCellEdge** und **SolidCellCenter** Zellen entsprechen den Festkörperzellen aus Kapitel 5.1. Sie sind beinahe identisch, haben lediglich andere geometrische Größen, welche darauf beruhen, dass die Bereiche der Platte verschieden diskretisiert werden können. Dabei handelt es sich bei der **SolidCellEdge** um die Zellen welche den oberen und unteren Randabschnitt der Platte beschreiben. Dadurch gibt es ein **SolidEdge1** und ein **SolidEdge2**. Das **SolidCenter** liegt jeweils immer zwischen zwei Kühlkanälen. Die Anzahl von **SolidCenter** Bereichen ist von der Anzahl der Durchläufe des Kühlkanals abhängig.

Durch einfache Berechnungen kann die gesamte Geometrie der Platte, sowie der einzelnen Zellen berechnet werden. So wird die Breite des **SolidEdge** Teils folgendermaßen bestimmt:

$$width_{edge} = (PlateWidth - n_{passes} d_{channel} - width_{center}) / 2 \quad (3.1)$$

Dabei bezieht sich  $width_{edge}$  auf die Breite eines Randabschnittes, von denen beide aber jeweils gleich groß sind. Die Angabe  $width_{center}$  bezieht sich auf den gesamten Bereich der zwischen den Kühlkanälen liegt und dessen Größe ist unabhängig von der Anzahl der Durchläufe des Kühlkanals.  $n_{SolidCenter}$  bezieht sich allerdings auf nur einen Abschnitt des Zentrums. Hat der Wärmeübertrager zum Beispiel 2 Kühlkanäle, gibt es nur ein Zentrum, dessen Breite auch  $width_{center}$  entspricht und  $n_{SolidCenter}$  Elemente beinhaltet. Hat der Wärmeübertrager 3 Kühlkanäle, gibt es 2 Zentren von denen beide zusammen die Breite  $width_{center}$  haben und es insgesamt 2 mal  $n_{SolidCenter}$  Elemente gibt.

Der Radius der runden **LiquidCellChangeOfStream** Zelle berechnet sich durch

$$R = 0,5 width_{center}. \quad (3.2)$$

Die Größen der einzelnen Zellen werden durch die gesamte Größe der Platte und der jeweiligen Diskretisierung berechnet. Zum Beispiel ermittelt sich die Länge  $x$  (siehe Abbildung 5.1) einer **SolidCellEdge** durch

$$x_{SCE} = width_{edge} / n_{SolidEdge} \quad (3.3)$$

und

$$y_{SCE} = PlateLength / n_{PlateLength}. \quad (3.4)$$

Die Größe  $z_{SCE}$  soll zur Vereinfachung gleich groß sein wie  $y_{SCE}$ , da nur zweidimensional diskretisiert wird.

Des Weiteren werden die Flächen für den Wärmeübergang und das Volumen der Zelle sowie der Umfang der horizontalen Fläche ermittelt. Für ein Quader ergibt sich dadurch

$$A_{SCExy} = x_{SCE} y_{SCE} \quad A_{SCExz} = x_{SCE} z_{SCE} \quad A_{SCEyz} = y_{SCE} z_{SCE} \quad (3.5)$$

$$V_{SCE} = x_{SCE} y_{SCE} z_{SCE} \quad (3.6)$$

und

$$U_{SCE} = 2 x_{SCE} + 2 y_{SCE}. \quad (3.7)$$

Die gleichen Berechnungen müssen auch für die **SolidCellCenter** aufgestellt werden. Dabei berechnet sich  $x_{SCC}$  durch die Abhängigkeit der Anzahl der Zentren von  $n_{passes}$  wie folgt:

$$x_{SCC} = width_{center} / (n_{SolidCenter} (n_{passes} - 1)) \quad (3.8)$$

Die weiteren Berechnungen erfolgen genauso wie bei der **SolidCellEdge**. Dabei sind die Größen  $y_{SCC}$  und  $z_{SCC}$  genauso groß wie beim **SolidCellEdge**.

Beim Flüssigelement entspricht  $d_{channel}$  dem Durchmesser der Außenwand des Kühlkanals. Der Durchmesser der Innenwand des Kanals berechnet sich durch:

$$d_{Liquid} = d_{channel} - 2 w_{Wall} \quad (3.9)$$

Die Bezeichnungen der Größen des Flüssigelements sind in Abbildung 5.2 zu sehen. Die Länge  $l$  einer **LiquidCell** entspricht der Länge  $y_{SCE}$  bzw.  $y_{SCC}$  eines Festkörperelements.

Die Innenflächen berechnen sich in Abhängigkeit des Winkels  $\mu$ .

$$A_{LCz} = 2 \frac{d_{Liquid}}{2} \sin\left(\frac{\mu}{2}\right) l \quad (3.10)$$

Wobei  $A_{LCz}$  der abgeflachten Fläche des Rohres am Flüssigelement entspricht und sich aus der Länge der Kreissehne multipliziert mit der Länge der Zelle errechnet. Für die beiden seitlichen gleichgroßen Flächen ergibt sich:

$$A_{LCy1,2} = \left( \pi \frac{d_{Liquid}}{2} - \frac{d_{Liquid}}{2} \frac{\mu}{2} \right) l \quad (3.11)$$

Für die Zelle der Rohrkrümmung (**LiquidCellChangeOfStream**), berechnen sich die verschiedenen Flächen des Wärmeübergangs in Abhängigkeit des Radius des Kreisabschnittes  $\delta$ , welcher in diesem Modell konstant bei  $180^\circ$  liegt, wie folgt:

$$A_{LCCOSz} = \frac{2}{3} \pi \frac{d_{Liquid}}{2} \left( R + \frac{d_{channel}}{2} \right) (2 \pi - \delta) \quad (3.12)$$

$$A_{LCCOSy1} = \frac{2}{3} \pi \frac{d_{Liquid}}{2} (R + w_{Wall}) (2 \pi - \delta) \quad (3.13)$$

$$A_{LCCOSy2} = \frac{2}{3} \pi \frac{d_{Liquid}}{2} (R + d_{channel} - w_{Wall}) (2 \pi - \delta) \quad (3.14)$$

$A_{LCCOSy1}$  entspricht der Fläche an der Innenseite des Krümmung,  $A_{LCCOSy2}$  der Außenseite und  $A_{LCCOSz}$  dem mittleren Radius des jeweils gedrittelten Torus (siehe Abb. 5.3). Für die Beschreibung der Wärmeleitpaste gibt es im **Base Record** die Parameter  $w_{grease}$ , der Dicke der Schicht, und  $\lambda_{grease}$ , der Wärmeleitfähigkeit der Paste.

Da es sich um ein **record** handelt werden alle Parameter lediglich mittels **parameter**, der Modelica Einheit und dem Namen des Parameters deklariert ohne irgendwelche Werte anzugeben.

## 3.2 Cold Plate Modell

Dieses **record** erbt durch den Befehl **extends Base Record** vom oben beschriebenen **Base Record**. Erst hier werden den Parametern Werte zugewiesen, wodurch alle Parameter durch die oben genannten Gleichungen berechnet werden können.

Durch die Vererbung und Erstellung eines **Base Records** können mehrere Wärmeübertrager Modelle mit verschiedenen Diskretisierungen oder Größen erstellt werden, welche durch die Funktion eines **replaceable record** im „Dropdown Menü“ des UserInterface einfach geändert werden können.

Hier stellt sich die Frage der sinnvollen Diskretisierung des Wärmeübertragers, welche im Kapitel 9 der Validierung behandelt wird.

## 4 Transportphänomene

Im Kapitel Transportphänomene werden alle Bewegungen und Interaktionen mit der Umgebung und dem Fluid zusammengefasst. Es beinhaltet die Konvektion zur Umgebung, den Wärmeübergang im Fluid und die Druckverluste im strömenden Medium. Die verschiedenen Phänomene sind jeweils ähnlich aufgebaut. Sie haben alle ein `partial model` auf dem alle weiteren implementierten Modelle aufgebaut sind und von diesem erben.

Durch ein `replaceable model` werden die Transportphänomene im Modell der Umgebung und der Flüssigzelle eingebunden, sodass die unterschiedlichen Modelle durch das „Dropdown Menü“ in der Benutzeroberfläche ausgewählt und geändert werden können.

### 4.1 Konvektion am Festkörperelement

Bei der Konvektion handelt es sich um einen makroskopischen Wärmetransport der ausschließlich in Fluiden stattfindet. Man unterscheidet zwischen erzwungener Konvektion, die beispielsweise durch Wind, einem strömendem Medium, verursacht wird, und der freien oder natürlichen Konvektion, welche durch die Entstehung eines natürlichen Auftriebs stattfindet [11].

Konvektion tritt nur an den Randzellen auf. Deshalb ist die Konvektion als externes Modell implementiert, welche die Wärmeübergangszahl  $\alpha$  der Umgebung und somit dem Festkörperelement übergibt.

Zur Auswahl stehen eine konstanter Wärmeübergang, bei dem  $\alpha$  konstant ist, sowie die freie Konvektion an der vertikalen und der horizontalen Platte und die erzwungene Konvektion an der senkrecht stehenden Platte.

Auch die Flüssigzelle, welche die Biegung des Rohres beschreibt, befindet sich außerhalb der „Cold Plate“ und ist nicht mit der Leistungselektronik verbunden, sondern in Kontakt mit der Umgebung. Da auch hier Konvektion auftritt, befindet sich das Modell der Konvektion am horizontalen Zylinder des Flüssigelements im Paket der Konvektion am Festkörperelement.

#### 4.1.1 Modell für Konvektion

Alle fünf Arten der implementierten Konvektion beruhen auf dem `partial model PartialHeatTransfer` wie es in Abbildung 4.1 zu sehen ist.

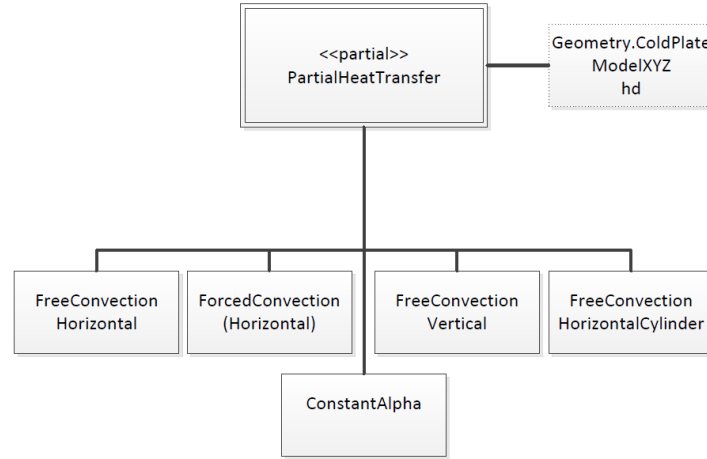


Abbildung 4.1: Struktogramm Konvektion am Festkörperelement

In diesem Modell werden die Variablen und Parameter als **inputs** definiert, welche für alle Arten der Konvektionen benötigt werden. **Output** dieses Modells ist die Wärmeübergangszahl  $\alpha$ . Durch die Deklaration der Parameter als Inputs müssen ihnen erst zu einem späteren Zeitpunkt Werte zugewiesen werden.

Die Geometriedaten der Platte werden als externes Paket eingebunden, siehe Abb. 4.1. Als weitere Eingaben werden zum einen die Zelltemperatur  $T_{Cell}$ , die charakteristische Länge  $L$ , sowie die Stoffgrößen der Umgebungsluft, die Temperatur  $T_{Ambient}$ , die spezifische Wärmekapazität  $c_p$ , die Dichte  $\rho$ , die dynamische und kinematische Viskosität  $\eta$  und  $\nu$ , die Wärmeleitfähigkeit  $\lambda$  und der isochore Spannungskoeffizient  $\beta$  benötigt.

Bei den Modellen der freien Konvektion an der vertikalen und horizontalen Platte, der erzwungenen Konvektion sowie der Konvektion am horizontalen Zylinder wird in den Modellen eine Nusselt- Zahl berechnet.

$$\alpha = Nu \frac{\lambda}{L} \quad (4.1)$$

Durch die Formel 4.1 wird schließlich  $\alpha$  berechnet.  $L$  ist die charakteristische Länge, welche für die verschiedenen Arten und Flächen der Konvektion anders ist und erst im Modell der Umgebung berechnet wird.

#### 4.1.2 Konstanter Wärmeübergang

Der Wärmeübergang mit einem vorgegebenen konstantem  $\alpha$  dient zur Überprüfung und zum Testen des Modells. Diese Funktion benötigt die Eingabe des Parameters *alphaConstant*. Durch

$$\alpha = alphaConstant \quad (4.2)$$

wird  $\alpha$  ausgegeben und kann dem Modell der Umgebung übergeben werden.

### 4.1.3 Freie Konvektion

Bei der freien Konvektion wird zwischen der Konvektion an der vertikalen und an der horizontalen Platte unterschieden. Wärmetransport an der horizontalen Platte findet an der Ober- und Unterseite der „Cold Plate“ statt. An den Seitenwänden des Wärmeübertragers findet der Wärmetransport an der vertikalen Platte statt.

Die Berechnung der Wärmeübergangszahl  $\alpha$  erfolgt durch sogenannte Nusselt- Korrelationen für welche wiederum die Grashofzahl  $Gr$ , die Prandtlzahl  $Pr$  und die Rayleighzahl  $Ra$  benötigt werden [12].

$$Gr = \beta | T_{Ambient} - T_{Cell} | \frac{g L^3}{\nu^2} \quad (4.3)$$

$$Pr = \frac{c_p \eta}{\lambda} \quad (4.4)$$

$$Ra = Gr Pr \quad (4.5)$$

#### 4.1.3.1 Vertikale Platte

Bei den Nusselt- Korrelationen muss zwischen laminarer und turbulenter Strömung unterschieden werden. Die Strömung ist laminar wenn  $Ra < 10^9$  ist. Ansonsten ist sie turbulent.

Für die Strömung gelten folgende Korrelationen für die mittlere Nusselt- Zahl [12]:

Laminar:

$$Nu = \frac{4}{3} 0,508 \sqrt{Pr} \sqrt[4]{\frac{Gr}{0,952 + Pr}} \quad (4.6)$$

Turbulent:

$$Nu = 0,135 \sqrt[3]{Gr Pr} \quad (4.7)$$

Die charakteristische Länge  $L$  bei der freien Konvektion an der senkrecht stehenden Platte entspricht der Höhe der Zelle, welche erst später im Modell der Umgebung berechnet wird und nur vollständigkeithalber hier genannt werden soll.

#### 4.1.3.2 Horizontale Platte

Für die freie Konvektion an der ebenen horizontalen Platte wurde der Fall „freie Konvektion an der Oberseite warme Platte/Unterseite kalte Platte“ gewählt. Der Wärmetransport findet mit der Umgebungsluft an der Oberseite des Elements statt, welche im Vergleich zur Unterseite, die in Kontakt mit der Kühlflüssigkeit ist, wärmer ist.

Für die Unterscheidung der laminaren oder turbulenten Strömung wird eine zusätzliche Variable  $\varphi(Pr)$  eingeführt.

$$\varphi(Pr) = (1 + (0,322/Pr)^{11/20})^{-20/11} \quad (4.8)$$



Somit gilt nach Baehr [13] für die laminare Strömung wenn  $Ra \cdot \varphi(Pr) < 7 \cdot 10^4$  und  $0 < Pr < \infty$  ist:

$$Nu = 0,766 Ra^{1/5} \varphi^{1/5} \quad (4.9)$$

Wenn  $Ra \cdot \varphi(Pr) \geq 7 \cdot 10^4$  dann ist folgende Nusselt- Korrelation für die turbulente Strömung zu verwenden:

$$Nu = 0,15 Ra^{1/3} \varphi^{1/3} \quad (4.10)$$

Die charakteristische Länge  $L$  entspricht dem Flächeninhalt geteilt durch den Umfang der überströmten Fläche.

#### 4.1.4 Erzwungene Konvektion

Durch Druckunterschiede entsteht eine Bewegung im Fluid wodurch die erzwungene Konvektion auftritt.

Betrachtet wird in dem Modell lediglich eine parallel angeströmte Platte.

Bei der erzwungenen Konvektion ist die Nusselt- Zahl nur von der Prandtlzahl und der Reynoldszahl abhängig. Die Reynoldszahl  $Re$  ist folgendermaßen definiert [14]

$$Re = \frac{v_\infty L}{\nu} \quad (4.11)$$

Die charakteristische Länge  $L$  entspricht der Länge der Platte über die das Fluid strömt.  $v_\infty$  ist die mittlere Geschwindigkeit des Fluides im Kanal.

Auch bei der erzwungenen Konvektion werden Korrelationen nach Baehr [15] verwendet:

$$Nu = \sqrt{Nu_{laminar}^2 + Nu_{turbulent}^2} \quad (4.12)$$

mit

$$Nu_{laminar} = \sqrt{\pi} \sqrt{Re} \varphi_{erzw} \quad (4.13)$$

und

$$Nu_{turbulent} = \frac{0,037 Re^{0,8} Pr}{1 + 2,443 Re^{-0,1} (Pr^{2/3} - 1)} \quad (4.14)$$

Diese Korrelation gilt im gesamten Bereich von  $10 < Re < 10^7$ . Bei turbulenter Strömung ist die Reynoldszahl groß, wodurch der laminare Nusselt- Term klein ist, und bei laminarer Strömung ist die Reynoldszahl klein, so dass der turbulente Nusselt- Term klein ist. Zur Berechnung von  $Nu_{laminar}$  wird zusätzlich  $\varphi_{erzw}(Pr)$  benötigt:

$$\varphi_{erzw}(Pr) = \frac{\sqrt{Pr}}{(1 + 2,55 Pr^{1/4} + 48,66 Pr)^{1/6}} \quad (4.15)$$

### 4.1.5 Freie Konvektion am horizontalen Zylinder

Wie bereits genannt, befindet sich die Flüssigzelle die die Biegung des Rohres beschreibt außerhalb des Wärmeübertragers wodurch an dieser Konvektion stattfindet. Hierfür wurde die freie Konvektion am horizontalen Zylinder nach [16] implementiert. Die charakteristische Länge  $L$  entspricht dabei dem Durchmesser des Zylinders.

$$Nu = C (Gr Pr)^n \quad (4.16)$$

Wobei  $C$  und  $n$  in Abhängigkeit der Rayleighzahl Tab. 4.1 bestimmt werden.

Ra	C	n
$Ra < 10^{-3}$	0,5	0
$10^{-3} < Ra < 5 \cdot 10^2$	1,18	1/8
$5 \cdot 10^2 < Ra < 2 \cdot 10^7$	0,54	1/4
$2 \cdot 10^7 < Ra < 10^{13}$	0,135	1/3

Tabelle 4.1: Parameter  $C$  und  $n$  für die freie Konvektion am horizontalen Zylinder

## 4.2 Konvektion der Rohrströmung

Dieses Modell beschreibt den Wärmeübergang in durchströmten Rohren. Es wird davon ausgegangen, dass die Strömung bei Eintritt in den Wärmeübertrager bereits hydrodynamisch ausgebildet ist und sich nicht mehr im Anlaufgebiet befindet. Auch hier wird zwischen laminarer und turbulenter Strömung unterschieden.

Wie bei der Konvektion am Festkörperelement besteht ein **partial model** von dem die beiden Wärmeübergänge „Konstanter Wärmeübergang“ und „Erzwungene Konvektion“ erben, siehe Abb. 4.2.

Ein Modell für eine freie Konvektion im Rohr, bei dem das Fluid nicht strömt, ist nicht notwendig, da das Fluid im Rohr des Wärmeübertrages stets in Bewegung ist.

### 4.2.1 Modell für Konvektion

Der **PartialHeatTransfer** benötigt als extern eingebundene Pakete die Geometriedaten der Platte und die Auswahl eines Kühlmediums als **replaceable package**. Weitere Eingaben die benötigt werden sind der Massenstrom  $\dot{m}$  sowie ein sogenannter *upstream state*. Durch diesen *upstream state* wird der Zustand des Mediums und somit die Fluidtemperatur bestimmt.

Der **output** dieses Modells ist die Wärmeübergangszahl  $\alpha$  welche nach Gln. 4.1 berechnet wird.

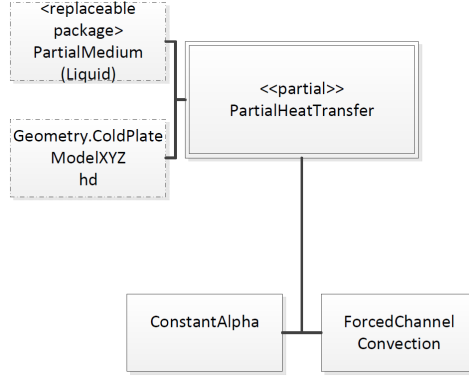


Abbildung 4.2: Struktogramm Konvektion der Rohrströmung

### 4.2.2 Konstanter Wärmeübergang

Der konstante Wärmeübergang der Rohrströmung ist ähnlich aufgebaut wie der konstante Wärmeübergang des Festkörperelements. Es wird der Parameter *alphaConstant* eingeführt und folglich ist:

$$\alpha = \alphaConstant \quad (4.17)$$

### 4.2.3 Erzwungene Konvektion

Für die Nusselt- Korrelationen der erzwungenen Konvektion werden die Korrelationen nach Gnielinski [17] verwendet. Da für das Modell eine elektronische Wärmequelle verwendet wird, welche einen konstanten Wärmestrom abgibt, werden die Korrelationen für konstante Wärmestromdichte verwendet.

Zunächst müssen die Stoffwerte des Fluides als Funktion des *upstream state* eingeführt werden. Dies erfolgt beispielsweise für die spezifische Wärmekapazität folgendermaßen:

$$Modelica.SIunits.SpecificHeatCapacity\ cpFluid = \\ MediumLiquid.specificHeatCapacityCp(upstreamState)$$

Ebenso werden auch die dynamische und kinematische Viskosität, die Dichte und die Wärmeleitfähigkeit des Fluides deklariert.

Für die Nusselt- Korrelationen werden die Reynoldszahl und die Prandtlzahl benötigt.

$$Re = \frac{\dot{m} d_{Liquid}}{\pi \left(\frac{d_{Liquid}}{2}\right)^2 \eta} \quad (4.18)$$

Anders wie bei der erzwungenen Konvektion am Festkörperelement wird in diesem Modell die Reynoldszahl in Abhängigkeit des Massenflusses  $\dot{m}$  anstelle der Geschwindigkeit

verwendet. Dies liegt daran, dass im Modell des Wärmeübertragers die Stärke des Kühlstromes durch den Massenfluss angegeben wird und nicht durch eine Geschwindigkeit. Ist die Strömung laminar, das heißt  $Re < 2300$  dann lautet die Nusselt- Korrelation:

$$Nu = [Nu_{m,q,1}^3 + 0,6^3 + (Nu_{m,q,2} - 0,6)^3]^{\frac{1}{3}} \quad (4.19)$$

mit

$$Nu_{m,q,1} = 4,364 \quad (4.20)$$

und

$$Nu_{m,q,2} = 1,953 \left( Re Pr \frac{d_{Liquid}}{l} \right)^{\frac{1}{3}} \quad (4.21)$$

Für die turbulente Strömung  $Re > 10^4$  ergibt sich die Nusselt- Korrelation zu

$$Nu = \frac{\frac{\xi}{8} Re Pr}{1 + 12,7 \sqrt{\left(\frac{\xi}{8}\right) (Pr^{\frac{2}{3}} - 1) \left(1 + \left(\frac{d_{Liquid}}{l}\right)^{\frac{2}{3}}\right)}}. \quad (4.22)$$

Dabei ist

$$\xi = (1,8 \log(Re) - 1,5)^{-2}. \quad (4.23)$$

Diese Korrelation gilt für den Reynoldsbereich von  $10^4 < Re < 10^6$  und  $0,1 < Pr < 1000$  sowie  $\frac{d_{Liquid}}{l} < 1$ .

Für größere Werte der Reynoldszahl und Abweichungen der Prandtlzahl sind keine weiteren Korrelationen implementiert. Die Werte eines Wärmeübertrages befinden sich im Normalfall in diesem Bereich. Es existieren lediglich Warnungen, welche dem Modellierer während der Simulation angezeigt werden, im Falle eines Über-/Unterschreitens des gültigen Wertebereichs.

Für den Übergangsbereich zwischen turbulenter und laminarer Strömung,  $2300 < Re < 10^4$ , gilt unter Berücksichtigung von  $0,6 < Pr < 1000$  und  $\frac{d_{Liquid}}{l} < 1$  folgende Nusselt-Korrelation:

$$Nu = (1 - \gamma) Nu_{L,2300} + \gamma Nu_{T,10000} \quad (4.24)$$

mit

$$\gamma = \frac{Re - 2300}{10^4 - 2300} \quad (4.25)$$

$Nu_{L,2300}$  ist dabei der Nusselt- Wert der sich nach der laminaren Nusselt- Korrelation (Gln. 4.19) bei einer Reynoldszahl von 2300 ergibt.  $Nu_{T,10000}$  ist der Nusselt- Wert bei einer turbulenten Strömung mit einer Reynoldszahl von 10000 (Gln. 4.22).

### 4.3 Druckverluste im Rohr

Durch die Wandreibung des Fluids im Rohr, der inneren Fluidreibung und der Rohrgeometrie geht Druck verloren. Durch die Änderung des Druckes, ändert sich auch der

Zustand des Fluides, da die Eigenschaften des Fluids von Druck, Enthalpie und dessen Massenanteil abhängen.

Die entstehende Druckdifferenz soll in diesem Modell berechnet werden. Dabei wird genauso wie bei den anderen Transportvorgängen von einem **partial model** ausgegangen, wie in Abbildung 4.3 zu sehen ist. Ausgehend von diesem existieren die Modelle ohne und konstante Druckverluste, sowie Korrelationen für Druckverluste bei laminarer beziehungsweise turbulenter Strömung im Rohr und für Rohre mit Änderung der Strömungsrichtung. Letztere wird für die Elemente der Rohrkrümmung verwendet.

### 4.3.1 Modell für Druckverluste

Der output dieses Modells ist der Druckverlust  $\Delta p$ . Wie auch in den anderen Paketen der Transportphänomene werden die Geometriedaten der Platte sowie die Auswahl eines Kühlmediums als **replaceable package** eingebunden. Außerdem wird wie beim **PartialHeatTransfer** ein *upstream state* sowie die Vorgabe eines Massenstroms benötigt.

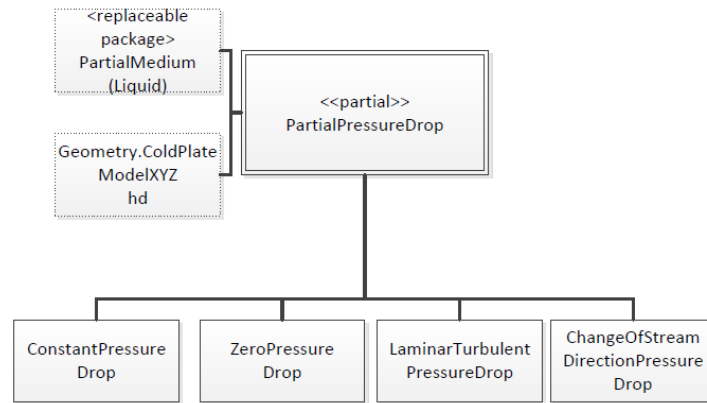


Abbildung 4.3: Struktogramm Druckverluste im Rohr

### 4.3.2 Ohne Druckverluste

Für das Modell ohne Druckverluste wird der Druckverlust Null.

$$\Delta p = 0 \quad (4.26)$$

### 4.3.3 Konstante Druckverluste

Das Modell der konstanten Druckverluste ist ähnlich aufgebaut wie das Modell des konstanten Wärmeübergangs des Festkörperelements und im Rohr.

Benötigt wird die Eingabe des Parameters *constantPressureDrop*. Daraus folgt, dass

$$\Delta p = \text{constantPressureDrop}. \quad (4.27)$$

#### 4.3.4 Druckverluste durch Rohrreibung

Bei der Berechnung des Druckverlustes in durchströmten Rohren wird von hydraulisch glatten Rohren ausgegangen. Diese Näherung gilt für Kupfer Rohre [18] welche bei Wärmeübertragern meist verwendet werden.

Die Stoffwerte der dynamischen Viskosität  $\eta$  und der Dichte  $\varrho$  des Fluides werden als Funktion des *upstream state* beschrieben:

$$\begin{aligned} \text{Modelica.SIunits.DynamicViscosity } \eta_{\text{Fluid}} &= \\ \text{MediumLiquid.dynamicViscosity}(\text{upstreamState}) & \\ \\ \text{Modelica.SIunits.Density } \rho_{\text{Fluid}} &= \\ \text{MediumLiquid.density}(\text{upstreamState}) & \end{aligned}$$

Für die Berechnung des Druckverlustes und zur Unterscheidung von laminarer und turbulenter Strömung wird die Reynoldszahl benötigt (siehe Gleichung 4.18).

Für die laminare Strömung, wenn  $Re < 2300$  ist, gilt folgende Korrelation [18]:

$$\Delta p = \frac{64}{Re} \frac{l}{d_{\text{Liquid}}} \frac{\varrho}{2} v^2 \quad (4.28)$$

Wobei die Geschwindigkeit  $v$  durch den Massenfluss ausgedrückt wird.

$$v = \frac{\dot{m}}{\varrho A} \quad (4.29)$$

$A$  entspricht der Kontaktfläche des Rohres. Diese ist bei einem Rohr:

$$A = \pi \left( \frac{d_{\text{Liquid}}}{2} \right)^2 \quad (4.30)$$

Für die turbulente Strömung,  $Re > 2300$ , ergeben sich nach Konakov Druckverluste von:

$$\Delta p = \xi_{\text{friction1}} \frac{l}{d_{\text{Liquid}}} \frac{\varrho}{2} \left( \frac{\dot{m}}{\varrho \pi \left( \frac{d_{\text{Liquid}}}{2} \right)^2} \right)^2 \quad (4.31)$$

mit

$$\xi_{\text{friction1}} = \frac{1}{(1,8 \log(Re) - 1,5)^2}. \quad (4.32)$$

Dabei ist  $\xi_{\text{friction1}}$  nach Konakov für den Reynoldszahlenbereich von  $2300 < Re < 10^7$  gültig [19]. Ist die Reynoldszahl größer, dann wird dem Modellierer eine Warnung ausgegeben.

### 4.3.5 Druckverluste für die Änderung der Strömungsrichtung

Durch die Krümmung des Rohres und der damit verbundenen Änderung der Geometrie gehen große Druckverluste einher. Zur Berechnung nach Idelchik [20] werden wie im Kapitel 4.3.4 die Reynoldszahl, die Dichte des Fluids  $\varrho$  und die dynamische Viskosität  $\eta$  benötigt.

Zudem müssen die beiden Parameter Radius  $R$  des Kurvenstückes und Länge des Kurvenabschnittes  $\delta$  in Grad angegeben werden.

Folglich berechnet sich der Druckverlust durch:

$$\Delta p = \frac{0,0175 \xi_{friction2} \delta \varrho R}{2 d_{Liquid}} \frac{\dot{m}}{\varrho \pi \left( \frac{d_{Liquid}}{2} \right)^2} \quad (4.33)$$

Die Größe von  $\xi_{el}$  wird jeweils für laminare, turbulente und dem Übergangsbereich verschieden bestimmt.

Ist  $Re \cdot \sqrt{\frac{d_{Liquid}}{2R}} < 600$  dann berechnet sich  $\xi_{el}$  durch:

$$\xi_{friction2} = \frac{20}{Re^{0,65}} \left( \frac{d_{Liquid}}{2R} \right)^{0,175} \quad (4.34)$$

Wenn  $Re \cdot \sqrt{\frac{d_{Liquid}}{2R}} < 1400$  ist, dann gilt

$$\xi_{friction2} = \frac{19,4}{Re^{0,55}} \left( \frac{d_{Liquid}}{2R} \right)^{0,225} \quad (4.35)$$

Für  $Re \cdot \sqrt{\frac{d_{Liquid}}{2R}} > 1400$  folgt

$$\xi_{friction2} = \frac{5}{Re^{0,45}} \left( \frac{d_{Liquid}}{2R} \right)^{0,275} \quad (4.36)$$

Dabei ist zu beachten, dass diese Korrelationen nur für den Bereich von  $50 < Re \cdot \sqrt{\frac{d_{Liquid}}{2R}} < 5000$  gelten. Wird dieser Bereich unter-/überschritten dann erscheint im Simulationsfenster eine Warnung.

# 5 Zellen

Der Wärmeübertrager wird in Flüssig- und Festkörperelemente eingeteilt. Dabei stellt ein Element eine nach dem Prinzip des Finite-Volumen Verfahrens diskretisierte Zelle dar. Letztendlich können diese Zellen beliebig miteinander verbunden werden und stellen so den Wärmeübertrager dar.

Innerhalb einer Zelle werden alle Ströme innerhalb des Kontrollvolumens aufsummiert und ergeben, im Fall des Festkörperelements, eine Zelltemperatur im Mittelpunkt des Volumens. Die Ströme bleiben stets erhalten und gehen nicht verloren. Das bedeutet, dass das was aus einer Zelle hinausströmt in die nächste hineinströmt. Dadurch kann der Wärmeübertrager verschieden groß diskretisiert werden, ohne dass sich die Ströme ändern. Ströme sind im Fall der Festkörperzelle lediglich Wärmeströme, im Fall der Flüssigzelle sowohl Wärmeströme als auch die Strömung des Kühlmediums.

Genauso wie bei den Transportphänomenen gibt es jeweils ein **partial model** für die Flüssig- und die Festkörperzelle, in der alle nötigen Parameter deklariert werden und die benötigten Pakete eingebunden werden, von der alle weiteren Zellen hierarchisch erben.

## 5.1 Festkörperelement

Durch die unterschiedliche Temperaturverteilung in der Zelle  $T_{Cell}$  und an der Außenseite der Zelle, findet Wärmeleitung statt. Hierbei handelt es sich um einen zwischenmolekularen Wärmetransport, bei dem durch atomare Wechselwirkungen Energie übertragen wird [11].

Es gibt zwei Arten von Zellen, die `SolidCellEdge`, welche den beiden Randbereichen des Wärmeübertragers entspricht und die `SolidCellCenter`, die den Bereich zwischen den Kühlkanälen diskretisiert.

### 5.1.1 Modell für Festkörperzelle

Ein Festkörperelement hat im Raum sechs Richtungen in der Wärme ausgetauscht werden kann. Ein solches Element kann man sich als einfachen Quader mit einer Länge, Breite und Höhe vorstellen, Abb. 5.1. An jeder Seite dieses Quaders befindet sich ein `HeatPort`, der aus der Modelica Standardbibliothek stammt.



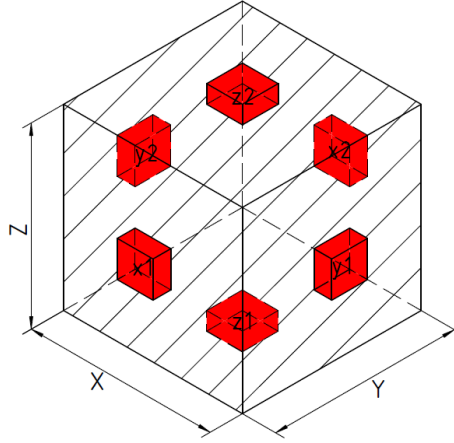


Abbildung 5.1: Festkörperelement mit HeatPorts

In der Festkörperzelle wird lediglich die Wärmeleitung durch das Material beschrieben. Die Konvektion der Randzellen zur Umgebung wird nicht in den Festkörperzellen implementiert, da es sonst zu viele verschiedene Zellen gäbe. Jede Zelle an einem anderen Randbereich hätte eine andere Wärmedurchgangszahl  $k$  am **HeatPort**. Deshalb wird die Konvektion zu einem späteren Zeitpunkt mit dem Modell der Umgebung modelliert.

$T_{start}$ , der Startwert der Zelltemperatur  $T_{Cell}$ , wird als Parameter eingegeben. Und  $T_{Cell}$  soll in dem Modell berechnet werden.

Das Material der Zelle wird durch ein **replaceable package** eingebunden. Hierbei werden die Materialien aus der Metallbibliothek, Kapitel 6, verwendet. Dazu müssen die Stoffwerte der Wärmeleitfähigkeit  $\lambda$ , der spezifischen Wärmekapazität  $c_p$  und der Dichte  $\rho$  durch die Interpolationsfunktion aus der **FluidDynamics** Bibliothek mittels der Zelltemperatur  $T_{Cell}$  interpoliert werden.

Um die Größe des Elements, also Länge, Breite und Höhe zu erhalten, muss das Geometrie Record eingebunden werden.

Außerdem wird für die Berechnung der Zelltemperatur nach dem Verfahren der Finiten-Volumen eine sogenannte Wärmekapazität  $C$  des Elements mit Volumen  $V$  benötigt:

$$C = c_p \rho V \quad (5.1)$$

### 5.1.2 Festkörperzelle: SolidCellEdge

Für jeden **HeatPort** wird die Wärmedurchgangszahl  $k$  und der Wärmestrom  $\dot{Q}$  einzeln berechnet. Somit ergibt sich aus der Grundgleichung:

$$k = \frac{\lambda}{l} \quad (5.2)$$

für den jeweiligen **HeatPort**:

$$k_{SCEx1} = \frac{\lambda}{y_{SCE}/2} \quad k_{SCEx2} = \frac{\lambda}{y_{SCE}/2} \quad k_{SCEy1} = \frac{\lambda}{x_{SCE}/2}$$

$$k_{SCEy2} = \frac{\lambda}{x_{SCE}/2} \quad k_{SCEz1} = \frac{\lambda}{z_{SCE}/2} \quad k_{SCEz2} = \frac{\lambda}{z_{SCE}/2}.$$

Die Länge  $l$  entspricht hierbei jeweils der Hälfte der Höhe  $z_{SCE}$ , Breite  $x_{SCE}$  oder Länge  $y_{SCE}$  des Elements, da die Zelltemperatur in der Mitte des Quaders berechnet wird. Der Wärmestrom ist nach [21] folgendermaßen definiert:

$$\dot{Q} = k (\text{heatPort}.T - T_{Cell}) A \quad (5.3)$$

Wobei  $\text{heatPort}.T$  die Temperatur an der Außenseite des Elements ist und durch den jeweiligen **HeatPort** berechnet wird.

$A$  entspricht der jeweiligen Wandfläche an der die Wärmeleitung stattfindet und  $k$  der entsprechenden Wärmedurchgangszahl am **HeatPort**. Dadurch ergibt sich:

$$\dot{Q}_{SCEx1} = k_{SCEx1} (\text{heatPort}_{SCEx1}.T - T_{Cell}) A_{SCExz} \quad (5.4)$$

$$\dot{Q}_{SCEx2} = k_{SCEx2} (\text{heatPort}_{SCEx2}.T - T_{Cell}) A_{SCExz} \quad (5.5)$$

$$\dot{Q}_{SCEy2} = k_{SCEy2} (\text{heatPort}_{SCEy1}.T - T_{Cell}) A_{SCEyz} \quad (5.6)$$

$$\dot{Q}_{SCEy2} = k_{SCEy2} (\text{heatPort}_{SCEy2}.T - T_{Cell}) A_{SCEyz} \quad (5.7)$$

$$\dot{Q}_{SCEz2} = k_{SCEz1} (\text{heatPort}_{SCEz1}.T - T_{Cell}) A_{SCExy} \quad (5.8)$$

$$\dot{Q}_{SCEz2} = k_{SCEz2} (\text{heatPort}_{SCEz2}.T - T_{Cell}) A_{SCExy} \quad (5.9)$$

Um das Festkörperelement vollständig zu deklarieren und die Zelltemperatur  $T_{Cell}$  berechnen zu können, müssen die Wärmeströme letztlich aufsummiert werden.

$$C \frac{dT_{Cell}}{dt} = \sum \dot{Q} \quad (5.10)$$

### 5.1.3 Festkörperzelle: SolidCellCenter

Die **SolidCellCenter** ist beinahe genauso aufgebaut wie die **SolidCellEdge** allein mit kleinen Unterschieden. Zum einen ist die Breite  $x_{SCC}$  durch die unterschiedliche Diskretisierung von Rand und Zentrum anders, und zum anderen befindet sich auf der Zentrumszelle im Zusammenbau des Wärmeübertragers die Elektronik. Zur besseren Übertragung der Wärme befindet sich zwischen Zelle und Elektronik eine Wärmeleitpaste. Deshalb ist die Wärmedurchgangszahl  $k$  an der Stelle  $z2$ , also an der oberen Seite der Zelle wie folgt:

$$k_{SCCz2} = \frac{1}{\frac{z_{SCC}}{2} \frac{1}{\lambda} + \frac{w_{grease}}{\lambda_{grease}}} \quad (5.11)$$

## 5.2 Flüssigelement

Die Flüssigzellen stellen den Kühlkanal des Wärmeübertragers dar, durch welchen die abgegebene Wärme der Leistungselektronik abtransportiert werden soll. Dadurch findet in diesen Zellen sowohl Wärmetransport, also Energietransport, als auch Massenfluss, also Teilchentransport statt.

Es gibt zwei verschiedene Arten von Zellen die sich in ihrer Geometrie unterscheiden. Die Zellen, welche sich in der Platte des Wärmeübertragers befinden, und solche welche außerhalb der Platte sind und die Änderung der Strömungsrichtung modellieren. An den äußeren Zellen kann durch das Modell der Umgebung eine Konvektion am horizontalen Zylinder stattfinden.

### 5.2.1 Modell für Flüssigzelle

Das Fluidelement hat zwei **FluidPorts** für den Ein- und Auslass des Kühlmittels und drei **HeatPorts** für die Wärmeleitung im Rohr.

Für das Kühlmedium im Kanal (*MediumLiquid*) wird aus der **AirSystemsMedia** Bibliothek das Kühlmedium und für die Rohrwand das Material aus der Metallbibliothek als **replaceable package** integriert. Dadurch kann die Wärmeleitfähigkeit  $\lambda$  durch die Temperatur am jeweiligen **HeatPort** wie beim Festkörperelement interpoliert werden.

Für die Wärmeleitung durch die Rohrströmung im Fluid und dem Druckverlust im Rohr werden die beiden Transportphänomene **ChannelHeatTransfer** und **ChannelPressure-Drop** als **replaceable model** in das partielle Modell integriert.

Als zusätzlicher **input** werden der Massenfluss  $\dot{m}$ , sowie der Parameter  $T_{start}$ , welcher der Startwert der Variable  $T_{Channel}$  ist, benötigt. Außerdem wird für die Berechnung und Beschreibung des Zustand des Fluids der *upstream state* benötigt. Dieser ist folgendermaßen deklariert:

$$\begin{aligned} upstream\ state = \\ MediumLiquid.setState_{phX}(fluidPort_{in}.p, inStream(fluidPort_{in}.h_{outflow}) + \\ dh/2, fluidPort_{in}.Xi_{outflow}) \end{aligned}$$

Wobei das  $dh$  eine weitere eingeführte Variable ist und die Enthalpieänderung des Fluides vom eingehenden zum ausgehenden **FluidPort** beschreibt. Auch dieses besitzt einen Startwert  $dh_{start}$ .

Die Funktion  $setState_{phX}$  stammt aus der **AirSystemsMedia** Bibliothek, welche mittels Eingabe von Druck  $p$ , welcher hier dem Druck am **FluidPort** entspricht, der Enthalpie, und dem Massenanteil am **FluidPort**, den Zustand und dadurch die Temperatur des Kühlmediums bestimmt. Da die Temperatur in der Mitte des Kanals zwischen Ein- und Ausgang des Kanals bestimmt werden soll, wird der Enthalpiewert so gewählt, dass vom einströmenden Port die Hälfte der Änderung der Enthalpie  $dh$  dazu addiert wird, so dass man sich im Mittelpunkt des Kanals befindet.

Der *inStream* Operator gibt den Wert der angenommenen Flussrichtung unabhängig von der tatsächlichen wieder [22].

Zusätzlich hat das Modell einen Pseudo- State. Dieser kann optional an- und ausgeschaltet werden. Der Pseudo- State benötigt zusätzlich die Deklaration einer Zeitkonstante *timeConst*, deren voreingestellter Wert bei 100s liegt. Ist der State an, dann findet zu Beginn kein Wärmestrom statt. Dadurch wird vermieden, dass ein großes System an nicht- linearen Gleichungen entsteht.

### 5.2.2 Flüssigzelle

Bei der Flüssigzelle, welche sich in der Platte befindet, ist der obere **HeatPort** an der abgeflachten Seite des Rohres mit der Elektronik verbunden. Die beiden seitlichen stehen in Kontakt mit den Festkörperelementen wie in Abbildung 5.2 zu sehen ist.

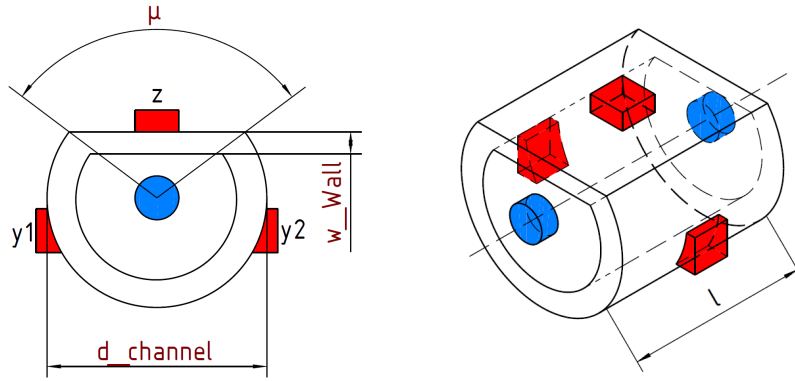


Abbildung 5.2: Flüssigelement mit **HeatPorts**(rot) und **FluidPorts**(blau)

Durch die unterschiedlichen Geometrien der Rohrwand ergeben sich verschiedene Wärmedurchgangszahlen. Der Wärmeübergang an der abgeflachten Seite des Rohres entspricht dem Wärmeübergang an einer ebenen Platte mit der Wärmeleitschicht:

$$k_{LCz} = \frac{1}{\frac{\frac{d_{channel}}{2} - \frac{d_{Liquid}}{2}}{\lambda} + \frac{1}{\alpha} + \frac{w_{grease}}{\lambda_{grease}}} \quad (5.12)$$

Der zusätzliche Term mit  $\alpha$  entspricht der Konvektion der Rohrströmung, dem **Channel-HeatTransfer**. Für die beiden seitlichen **HeatPorts** muss der logarithmische Temperaturverlauf durch die Rohrwand mitberücksichtigt werden. Damit bildet sich die Wärmedurchgangszahl zu:

$$k_{LCy1,2} = \frac{1}{\frac{d_{Liquid}}{2} \left( \frac{l}{\lambda} \ln \left( \frac{\frac{d_{channel}}{2}}{\frac{d_{Liquid}}{2}} \right) + \frac{1}{\alpha \frac{d_{Liquid}}{2}} \right)} \quad (5.13)$$

Unter Berücksichtigung der unterschiedlichen Flächen  $A$ , welche im **Base Record** berechnet werden, bildet sich der Wärmeübergang des jeweiligen Ports zu:

$$\dot{Q}_{LCy1} = k_{LCy1} (\text{heatPort}_{LCy1} \cdot T - T_{Channel}) A_{LCy1}. \quad (5.14)$$

$$\dot{Q}_{LCy2} = k_{LCy2} (\text{heatPort}_{LCy2} \cdot T - T_{Channel}) A_{LCy2}. \quad (5.15)$$

$$\dot{Q}_{LCz} = k_{LCz} (\text{heatPort}_{LCz} \cdot T - T_{Channel}) A_{LCz}. \quad (5.16)$$

Des Weiteren müssen für das Fluid Erhaltungsgleichungen beschrieben werden. Die **FluidPorts** benötigen Eingaben für den Massenfluss, den Druck, der Enthalpie und dem Massenanteil. Die Massenerhaltung kann folgendermaßen beschrieben werden:

$$0 = \text{fluidPort}_{in} \cdot \dot{m} + \text{fluidPort}_{out} \cdot \dot{m} \quad (5.17)$$

Somit ist der Massenfluss der in die Zelle hinein fließt der Gleiche wie er austritt. Folglich geht keine Masse verloren und die Massenerhaltung ist erfüllt.

Dadurch, dass die Ein- und Austrittsfläche des Flüssigelements gleich groß sind, kann die Impulserhaltung wie folgt formuliert werden:

$$\Delta p = \text{fluidPort}_{in} \cdot p - \text{fluidPort}_{out} \cdot p \quad (5.18)$$

Wobei die Größe des  $\Delta p$  durch das Modell des Druckverlustes berechnet wird.

Für den Massenanteil und der Enthalpie werden die nachfolgenden Formulierungen getroffen:

$$\text{fluidPort}_{in} \cdot X_{i_{outflow}} = \text{inStream}(\text{fluidPort}_{out} \cdot X_{i_{outflow}}) \quad (5.19)$$

$$\text{fluidPort}_{out} \cdot X_{i_{outflow}} = \text{inStream}(\text{fluidPort}_{in} \cdot X_{i_{outflow}}) \quad (5.20)$$

$$\text{fluidPort}_{out} \cdot h_{outflow} = \text{inStream}(\text{fluidPort}_{in} \cdot h_{outflow}) + dh \quad (5.21)$$

$$\text{fluidPort}_{in} \cdot h_{outflow} = \text{fluidPort}_{out} \cdot h_{outflow} - dh \quad (5.22)$$

Der Massenanteil der ausströmt ist somit der Gleiche wie er hineinströmt. Die Enthalpie am Ausgang des **FluidPorts** nimmt durch die Temperaturänderung um den Anteil  $dh$  zu. Bei der nächsten Zelle entspricht die Enthalpie am Eingang des Ports, der am Ausgang der davor liegenden Zelle.

Ist der Pseudo- State aktiviert, dann ergibt sich die Energieerhaltung zu:

$$\frac{dh}{dT} \text{timeConst} = (\dot{Q}_{LCy1} + \dot{Q}_{LCy2} + \dot{Q}_{LCz}) / \text{fluidPort}_{in} \cdot \dot{m} - dh \quad (5.23)$$

Ist er nicht aktiviert dann folgt:

$$dh \text{fluidPort}_{in} \cdot \dot{m} = \dot{Q}_{LCy1} + \dot{Q}_{LCy2} + \dot{Q}_{LCz} \quad (5.24)$$

Da

$$dh = c_p dT \quad (5.25)$$

gilt, wird durch das  $dh$  die Temperaturänderung des Fluides berechnet, was durch den *upstream state* geschieht. Dabei entspricht  $T_{Channel}$  dem *upstreamState.T*.

### 5.2.3 Flüssigzelle der Rohrkrümmung

Da die flüssige Zelle in der Rohrkrümmung eine andere Geometrie und somit andere Korrelationen der Wärmeleitung hat, muss es hierfür eine weitere Zelle, die `LiquidCell-ChangeOfStream` geben.

Anders als die normale Flüssigzelle ist das Rohr an der Oberseite nicht abgeflacht. Dadurch ergeben sich ausschließlich Wärmeübergangskoeffizienten am Zylinder.

Somit ändert sich lediglich die Wärmedurchgangszahl am `HeatPortLCCOSz`

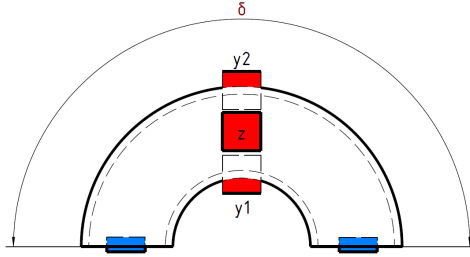


Abbildung 5.3: Flüssigzelle der Rohrkrümmung mit den `HeatPorts`(rot) und `FluidPorts`(blau)

$$k_{LCCOSz} = \frac{1}{\frac{d_{Liquid}}{2} \left( \frac{l}{\lambda} \ln \left( \frac{\frac{d_{channel}}{2}}{\frac{d_{Liquid}}{2}} \right) + \frac{1}{\alpha \frac{d_{Liquid}}{2}} \right)} \quad (5.26)$$

Falls eine Konvektion an diesem Rohrstück simuliert werden soll, geschieht dies anhand des Modells der Umgebung.

## 6 Metall Materialbibliothek

Für eine realitätsnahe Simulation des Wärmeübertragers ist es notwendig, nicht nur das Kühlmittel und die Umgebungsluft als Medien zu verwenden und somit die genauen Stoffwerte bei gegebener Temperatur zu verwenden, sondern auch das Material des Festkörperelements und des Kühlrohres aus Tabellenwerten zu interpolieren.

Als Stoffwerte werden für dieses Modell lediglich die Dichte  $\rho$ , die spezifische Wärmekapazität  $c_p$  und die spezifische Wärmeleitfähigkeit  $\lambda$  benötigt.

Die Metall Materialbibliothek beruht wie alle Medien in Modelica auf dem **Partial-Medium**. Da die Stoffwerte von Metall nur von der Temperatur abhängen, genügt es die oben genannten Stoffwerte in tabellarischer Form in Abhängigkeit der Temperatur zu speichern. Hierzu wird das **TableBased** Paket aus der Modelica Standardbibliothek verwendet.

Im Modell können diese Werte durch die Funktion „Interpolieren ohne zu Extrapolieren“, welche aus der **FluidDynamics** Bibliothek stammt, aufgerufen werden. Diese Funktion benötigt als Eingaben die Temperatur und den Druck. Da die Stoffwerte von Metall nur in Abhängigkeit der Temperatur implementiert sind, kann für den Druck jeglicher Wert angegeben werden.

Bei Einbindung der Metall Materialbibliothek stehen die folgenden Elemente und Legierungen zur Auswahl. Die tabellarischen Werte der Materialien sind im Anhang zu finden und stammen aus [23].

- Aluminium und Aluminiumlegierungen
  - Reines Aluminium
  - Reines Aluminium mit  $c_p = 0$
  - Reines Aluminium mit  $c_p = 0$  und  $\lambda = \textit{konstant}$
  - Aluminium 2024T6
  - Aluminium 195 Guss
- Kupfer und Kupferlegierungen
  - Reines Kupfer
  - Reines Kupfer mit  $c_p = 0$
  - Reines Kupfer mit  $c_p = 0$  und  $\lambda = \textit{konstant}$
  - Kommerzielle Bronze
  - Phosphor Bronze
  - Messing
  - Konstantan
- Eisen und Eisenlegierungen

- Reines Eisen
- Armco
- Reines Silber
- Stahl und Stahllegierungen
  - gewöhnlicher Stahl
  - AISI 1010
  - Kohlenstoff Silizium
  - Kohlenstoff Mangan Silizium
- Edelstahllegierungen
  - AISI 302
  - AISI 304
  - AISI 316
  - AISI 347
- Silizium

Meist bestehen die Festkörper von Wärmeübertragern aus Aluminium, wodurch „reines Aluminium“ die Standard Vorgabe im Modell des Festkörperelements ist. Bei den Flüssigelementen bestehen die Kühlrohre meist aus Kupfer, folglich ist „reines Kupfer“ die Standard Vorgabe, siehe Tab. 6.1 und Tab. 6.2.

Um direkt den stationären Zustand im Wärmeübertragungsprozess zu erreichen, ist das fiktive Material „reines Aluminium und reines Kupfer mit  $c_p = 0$ “ in der Bibliothek vorhanden. Damit Dymola mit diesem Wert rechnen kann, ist  $c_p$  sehr klein gewählt, siehe Tab. 6.3, so dass es beinahe Null ist.

Zusätzlich sind noch zwei weitere fiktive Materialien in der Metallbibliothek vorhanden. Diese entsprechen den beiden Materialien mit  $c_p = 0$ , nur dass zusätzlich die spezifische Wärmeleitfähigkeit  $\lambda$  konstant ist. Mit diesen beiden Materialien kann das Temperaturverhalten besser betrachtet werden, da zum einen der stationäre Endzustand sofort erreicht wird, und zum anderen bei konstantem  $\lambda$  alle Elemente die gleiche Wärmeleitfähigkeit besitzen.

$T [K]$	100	200	300	400	600	800
$\varrho [kg/m^3]$			2702			
$c_p [J/kgK]$	482	798		949	1033	1146
$\lambda [W/mK]$	302	237		240	321	218

Tabelle 6.1: Stoffwerte von reinem Aluminium

$T [K]$	100	200	300	400	600	800	1000	1200
$\varrho [kg/m^3]$			8933					
$c_p [J/kgK]$	252	356		397	417	433	451	480
$\lambda [W/mK]$	482	413		393	397	366	352	339

Tabelle 6.2: Stoffwerte von reinem Kupfer



	Aluminium	Kupfer
$\varrho \text{ [kg/m}^3\text{]}$	2702	8933
$c_p \text{ [J/kgK]}$	$10^{-10}$	$10^{-10}$
$\lambda \text{ [W/mK]}$	240	400

Tabelle 6.3: Stoffwerte von reinem Aluminium und Kupfer mit  $c_p = 0$  und  $\lambda = konst$

# 7 Umgebung

Die Aufgabe dieses Paketes ist es, zum einen realistische Umgebungsbedingungen der „Cold Plate“ zu schaffen und zum anderen ein Modell zu haben, das die Konvektion an den verschiedenen Seiten der Zelle modelliert. Letztlich müssen diese beiden Modelle miteinander verbunden werden, damit die Konvektion mit der Umgebung im Modell des diskretisierten Wärmeübertragers eingebaut werden kann.

Abb. 7.1 zeigt wie die Umgebung aufgebaut ist. In der `AmbientCondition` stehen die Stoffwerte in tabellarischer Form. Durch Einbindung des `record ambient`, werden Temperatur und Dampfgehalt übergeben und die Stoffwerte können interpoliert werden. Zusammen mit dem Modell der `Convection`, in welchem das Transportphänomen Konvektion am Festkörperelement, `AmbientHeatTransfer` eingebunden ist, bilden diese das Modell `ConvectionAmbient`.

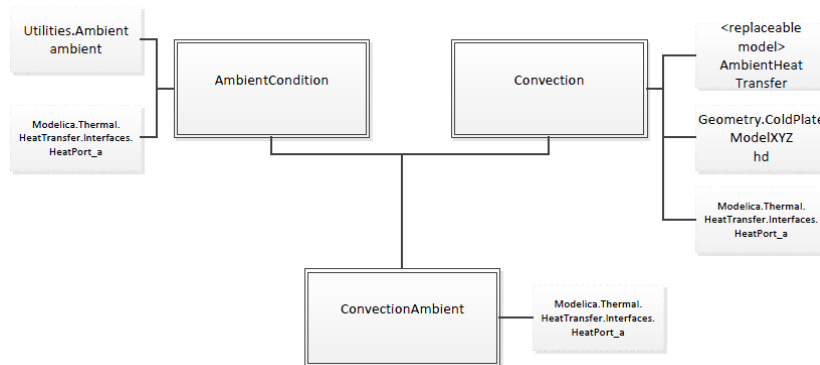


Abbildung 7.1: Struktogramm Umgebung

## 7.1 Base Record

Das `Base Record` ist ähnlich aufgebaut wie das `Geometrie Record`. Im `BaseRecord` werden die Parameter Temperatur  $T$  und Dampfgehalt  $\chi$  aufgelistet.

Erst im `record ambient` werden den Parametern Werte zugewiesen.

## 7.2 Stoffwerte der Umgebung

In diesem Modell (`AmbientCondition`) liegen die Stoffwerte der Dichte  $\rho$ , des isochoren Spannungskoeffizienten  $\beta$ , der spezifischen Wärmeleitfähigkeit  $\lambda$ , der kinematischen Viskosität  $\nu$ , der spezifischen Wärmekapazität von trockener Luft  $c_{pDry}$  und der von Wasserdampf  $c_{pVapour}$  in tabellarischer Form in Abhängigkeit der Temperatur vor (Tab. 7.1 [24] und 7.2 [25]).

Als `input` hat dieses Modell das `record ambient`, wodurch durch Übergabe von Temperatur und Dampfgehalt entsprechende Werte ermittelt werden können.

Der Druck in der Umgebung eines Wärmeübertragers liegt meist bei Normaldruck. Deshalb gelten die Werte bei 1 bar wodurch eine zusätzliche Abhängigkeit vom Druck in dem Modell dieser Arbeit unnötig wäre.

Die Werte können durch die Angabe einer Temperatur durch die Funktion „Interpolieren ohne zu Extrapolieren“ aus der `FluidDynamics` Bibliothek aufgerufen werden.

Die dynamische Viskosität  $\eta$  berechnet sich aus der kinematischen Viskosität und der Dichte:

$$\eta = \nu \rho \quad (7.1)$$

Da kinematische Viskosität und Dichte zur passenden Temperatur interpoliert werden, ist durch die Formel 7.1 auch die dynamische Viskosität zur Temperatur interpoliert.

$T[K]$	$\rho[kg/m^3]$	$\beta[1/K]$	$\lambda[W/mK]$	$\nu[10^{-7} m^2/s]$
243,15	1,434	4,133	22,02	109,4
253,15	1,377	3,967	22,81	117,7
263,15	1,325	3,815	23,59	126,2
273,15	1,276	3,674	24,36	135
283,15	1,231	3,543	25,12	144
293,15	1,189	3,421	25,87	153,2
303,15	1,149	3,307	26,62	162,6
313,15	1,112	3,201	27,35	173,2
323,15	1,078	3,101	28,08	182,2
333,15	1,046	3,007	28,80	192,2
343,15	1,015	2,919	29,52	202,5
353,15	0,9862	2,836	30,22	213
363,15	0,9590	2,758	30,93	223,7
373,15	0,9333	2,683	31,62	234,6
393,15	0,8857	2,546	32,99	257

Tabelle 7.1: Stoffwerte der Umgebung

Da die Änderung der spezifischen Wärmekapazität bei feuchter Luft relativ groß ist, wird diese durch die spezifische Wärmekapazität von trockener Luft und Wasserdampf

berechnet. Als weitere Angabe wird dafür der Dampfgehalt  $\chi$  benötigt. Die interpolierten Werte der spezifischen Wärmekapazität bei trockener Luft und bei Wasserdampf werden durch die Eingabe des Dampfgehaltes durch die folgende Formel berechnet [26]:

$$c_p = \frac{c_{pDry} + \chi c_{pVapour}}{1 + \chi} \quad (7.2)$$

Temperatur [K]	$c_{pDry}$ [J/kgK]	$c_{pVapour}$ [J/kgK]
253,15	1,0055	1,857
273,15	1,0056	1,858
293,15	1,0058	1,860
313,15	1,0062	1,863
333,15	1,0065	1,865
353,15	1,0071	1,868
373,15	1,0077	1,872
393,15	1,0085	1,875

Tabelle 7.2: Stoffwerte von  $c_{pDry}$  und  $c_{pVapour}$

Um das Modell der Stoffwerte der Umgebung mit der Konvektion verbinden zu können, benötigt das Modell der Stoffwerte der Umgebung einen **HeatPort**, den **Ambientport**.

## 7.3 Konvektion

Das Modell der Konvektion modelliert die Konvektion an der Festkörperzelle, also der **SolidCellEdge** und **SolidCellCenter**, und der **LiquidCellChangeOfStream**.

Die Konvektion stellt sozusagen die Verbindung zwischen der Umgebung und dem Festkörperelement dar. Dafür benötigt das Modell zwei **HeatPorts**. Einen der zur Umgebung führt, **AmbientConvectionPort**, und einen zum Festkörperelement, **SolidConvectionPort**. Um den Wärmeübergangskoeffizienten  $\alpha$  einzubinden, werden die Modelle für die Konvektion am Festkörperelement aus dem Kapitel Transportphänomene durch ein **replaceable package** implementiert.

Da das Transportphänomen **AmbientHeatTransfer** Eingaben für die Werte  $c_p$ ,  $\lambda$ ,  $\varrho$ ,  $\eta$ ,  $\nu$  und  $\beta$  benötigt, werden diese Werte erneut als **input** definiert und dem Transportphänomen übergeben. Die Temperatur  $T_{Ambient}$ , des **AmbientHeatTransfers** wird mit der Temperatur des **AmbientConvectionPort.T** gleichgesetzt. Gleicherweise entspricht die Temperatur  $T_{Cell}$  der Temperatur **SolidConvectionPort.T** des zweiten **HeatPorts**. Das Modell soll so funktionieren, dass durch einfache Eingabe in der Benutzeroberfläche bestimmt wird, an welcher Stelle die Konvektion stattfindet. Die Unterscheidung ist durch *if*-Bedingungen implementiert.

Da sich die charakteristische Länge für den Wärmeübergang und die Fläche an der Konvektion stattfindet ändert, je nachdem an welcher Zelle und Seite der Zelle die Konvektion stattfindet, ist es nötig einen Parameter *WhichCell* einzuführen, durch den festgelegt wird ob die Konvektion an der **SolidCellEdge**, **SolidCellCenter** oder an der **LiquidCellChangeOfStream** stattfindet. Zudem muss, im Falle der beiden Festkörperzellen unterschieden werden, ob die Konvektion an der Horizontalen oder Vertikalen stattfindet und da sich die charakteristische Länge bei der freien und erzwungenen horizontalen Konvektion ändert, muss auch hier differenziert werden.

Allgemein gilt für den Wärmestrom:

$$\dot{Q} = \alpha \Delta T A \quad (7.3)$$

Dabei ist  $\Delta T$  der Temperaturunterschied zwischen den beiden Ports **AmbientConvectionPort** und **SolidConvectionPort**.

Für die **SolidCellEdge** ergeben sich folgende Größen: Ist die Konvektion an der Horizontalen, ist die Wärmeübergangsfläche  $A_{SCExy}$ . Handelt es sich um eine freie Konvektion an der Horizontalen, dann beträgt die charakteristische Länge  $A_{SCExy}/U_{SCE}$ , ist sie erzwungen, ist die charakteristische Länge  $y_{SCE}$ .

Bei der vertikalen Konvektion muss zwischen den Seiten unterschieden werden, an der das Festkörperelement mit der Umgebung in Kontakt ist. Dabei ändern sich je nach Seite die Wärmeübergangsfläche zu  $A_{SCExz}$  oder  $A_{SCEyz}$ . Die charakteristische Länge beträgt im Fall der vertikalen Konvektion die Höhe  $z_{SCE}$ .

Die Größen der **SolidCellCenter** entsprechen denen der Randzelle, nur dass zum Beispiel anstelle von  $z_{SCE}$ ,  $z_{SCC}$  aus dem Geometrie Record gewählt werden muss.

Bei der Flüssigzelle entspricht die charakteristische Länge  $L$  dem Durchmesser des Kühlkanals  $d_{channel}$ .

Es muss unterschieden werden, an welcher Seite des Krümmungsbogens die Konvektion auftritt. Dazu sind entsprechende Größen im Geometrie Record enthalten,  $A_{LCCOSy1}$ ,  $A_{LCCOSy2}$  und  $A_{LCCOSz}$ .

## 7.4 Konvektion Umgebung

Das Modell der Konvektion der Umgebung verbindet, wie in Abb. 7.1 ersichtlich, die beiden Modelle **AmbientCondition** und **Convection**. Der **Ambientport**, der Stoffwerte der Umgebung, wird durch den Befehl *connect* mit dem **AmbientConvectionPort** der Konvektion verbunden. Die Werte  $cp_{Ambient}$ ,  $\lambda_{Ambient}$  usw. des **AmbientHeatTransfers**, bzw. des Modells der Konvektion, werden den entsprechenden interpolierten Werten der Stoffwerte der Umgebung zugewiesen.

Zusätzlich benötigt dieses Modell einen **HeatPort**, den **SolidPort**, welcher mit dem **SolidConvectionPort** der Konvektion verbunden wird. Mit diesem Port werden im diskretisierten Modell des Wärmeübertragers die **HeatPorts** der Zellen durch den *connect* Befehl mit der Konvektion der Umgebung verbunden.

## 8 Diskretisierung des Wärmeübertragers

Die Diskretisierung des Wärmeübertragers erfolgt durch die Deklaration der Zellen als Arrays, um dann alle einzelnen Elemente miteinander zum Modell des Wärmeübertrages zu verbinden.

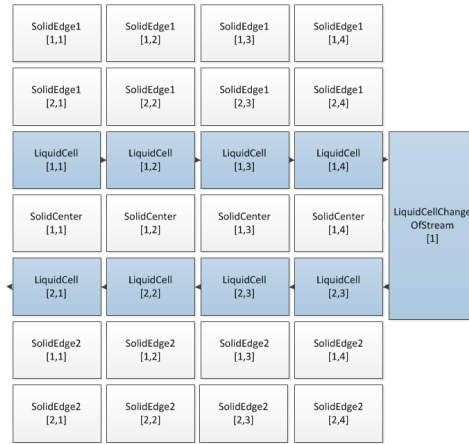


Abbildung 8.1: Beispiel eines diskretisierten Wärmeübertragers mit  $n_{PlateLength} = 4$ ,  $n_{passes} = 2$ ,  $n_{SolidEdge} = 2$  und  $n_{SolidCenter} = 1$

Der Zusammenbau der einzelnen Elemente geschieht durch eine Verschaltung von *for*-Schleifen und *if*-Bedingungen. Wobei die verschiedenen Arten der Zellen als Matrizen definiert werden und die einzelnen Matrizenelemente miteinander verbunden werden. Die Zellen werden dabei als **replaceable models** im diskretisierten Wärmeübertrager eingebunden.

Die Modelle für die Transportphänomene sowie die Materialien des Kühlmediums, des Kühlrohrs und des Festkörperelements werden ebenfalls durch **replaceable models** eingefügt und den jeweiligen Zellen übergeben. Durch die erneute Einbindung dieser Modelle im hierarchisch obersten Modell, können die auswechselbaren Modelle, wie zum Beispiel die Materialien, im UserInterface einfach geändert werden.

So existiert die Matrix  $solidCellEdge1[n_{SolidEdge}, n_{PlateLength}]$  welche den oberen Randbereich der „Cold Plate“ diskretisiert und die  $solidCellEdge2[n_{SolidEdge}, n_{PlateLength}]$  Matrix, welchen den unteren Bereich des Wärmeübertragers darstellt. Wobei der erste Eintrag

der Matrix die Zeile, der zweite die Spalte darstellt. Im Beispiel Abb. 8.1 ist *solidCellEdge1[2,4]*. Der untere Randbereich entspricht bei der Deklaration der Matrix der Oberen. Das Zentrum wird durch *solidCellCenter[nSolidCenter (n<sub>passes</sub> - 1), nPlateLength]* beschrieben. Die Größe des Zentrums, ist dabei von der Anzahl der Durchläufe des Kühlkanals durch die „Cold Plate“ abhängig. Im Beispiel, bei  $n_{passes} = 2$  ist *solidCellCenter[1,4]*. Auch der Kühlkanal wird durch eine Matrix aus **LiquidCells** gebildet. Dabei ist *liquidCell[n<sub>passes</sub>, nPlateLength]*, was sich im Beispiel Abb. 8.1 zu *liquidCell[2,4]* ergibt. Nach der Deklaration der vier Matrizen ist ersichtlich, dass alle verschiedenen Arten von Zellen in die Länge, also der Anzahl der Spalten, gleich diskretisiert sind mit  $n_{PlateLength}$ . Da sich die **LiquidCellChangeOfStream** außerhalb der Platte befindet und damit nicht in die Länge diskretisiert werden muss, ist sie als Vektor implementiert, *liquidCellChangeOfStream[n<sub>passes</sub> - 1]*. Dabei entspricht eine Zelle einem Rundbogen, welcher zwei Kühlkanäle miteinander verbindet. Im Beispiel Abb. 8.1 ergibt sich dadurch *liquidCellChangeOfStream[1]*.

Um die Randzellen mit der Konvektion der Umgebung verbinden zu können, benötigt dieses Modell insgesamt acht **HeatPorts**. Drei für die **SolidCellEdge** Elemente, für die Flächen  $A_{SCExy}$ ,  $A_{SCExz}$  und  $A_{SCEyz}$ . Bei den **SolidCellCenter** Elementen für die beiden Flächen  $A_{SCCxy}$  und  $A_{SCCxz}$ . Da die Zentrumszelle an der Fläche  $A_{SCCyz}$  mit dem Kühlkanal verbunden ist, findet dort keine Konvektion statt. Die **LiquidCellChangeOfStream** benötigt für alle drei Seiten, an denen jeweils ein **HeatPort** ist, einen Port für die Konvektion.

Für die Modellierung des Kühlkanals werden Ein- und Ausgang benötigt. Dafür werden aus der **FluidDynamics** Bibliothek **liquidInlet** und **liquidOutlet** verwendet. Dabei wird der **FluidPort** der ersten Zelle mit dem Inlet und der **FluidPort** der letzten Flüssigzelle mit dem Outlet verbunden. Dies geschieht durch:

---

```

1 connect(liquidInlet , liquidCell [1 , 1]. fluidPort_in );
2 if mod(hd.n_passes , 2) == 0 then
3   connect(liquidCell [hd.n_passes , 1]. fluidPort_out , liquidOutlet );
4 else
5   connect(liquidCell [hd.n_passes , hd.nPlateLength ]. fluidPort_out ,
6           liquidOutlet );
6 end if ;

```

---

Abbildung 8.2: Code: Verbindung LiquidCell mit liquidIn- Outlet

Wobei die Geometriegrößen des Wärmeübertragers durch *hd.Größe* aufgerufen werden. Da sich der Ausgang des Kühlmediums, je nachdem ob die Platte eine gerade oder ungerade Anzahl von  $n_{passes}$  hat, an der letzten Zelle in der ersten Spalte oder an der letzten Zelle in der letzten Spalte befindet, wird hier eine Unterscheidung durch den *modulo* Operator gemacht.

Wenn  $n_{passes}$  ohne Rest durch 2 teilbar ist, also gerade ist, dann befindet sich der Ausgang in der ersten Spalte, ansonsten in der letzten. Die Verbindung der Zellen mit dem Ein-

und Ausgang geschieht durch den Befehl *connect*.

Die Verbindung der Flüssigzellen untereinander geschieht durch folgenden Abschnitt des Codes:

---

```

1  for i in 1:2:hd.n_passes loop
2    for j in 1:hd.nPlateLength - 1 loop
3      connect(liquidCell[i,j].fluidPort_out,liquidCell[i,j + 1].
        fluidPort_in);
4    end for;
5  end for;
6  for i in 2:2:hd.n_passes loop
7    for j in 1:hd.nPlateLength - 1 loop
8      connect(liquidCell[i,j].fluidPort_in,liquidCell[i,j + 1].
        fluidPort_out);
9    end for;
10 end for;

```

---

Abbildung 8.3: Code: Verbindung der Flüssigzellen

In dem Code 8.3 laufen die *for*- Schleifen in der Zeilenanzahl in zweier Schritten, da bei der Betrachtung einer „Cold Plate“ von oben (siehe Abb. 8.1), bei einem ungeraden Kanal das Kühlmittel von links nach rechts, und bei einem geraden Kanal das Kühlmedium von rechts nach links läuft. Deshalb werden in diesem Fall die Zellen „falsch“ herum miteinander verbunden. Also der  $\text{FluidPort}_{in}$  der ersten Zelle mit dem  $\text{FluidPort}_{out}$  der nächsten Zelle.

Auch bei der Verbindung der *LiquidCell* mit der *LiquidCellChangeOfStream* muss darauf geachtet werden, ob es sich um einen geraden oder ungeraden Kühlkanal handelt, da sonst die Rundbogenzelle an der linken oder rechten Seite der Platte liegt:

---

```

1  for i in 1:2:hd.n_passes - 1 loop
2    connect(liquidCell[i,hd.nPlateLength].fluidPort_out,
        liquidCellChangeOfStreamDirection[i].fluidPort_in);
3    connect(liquidCellChangeOfStreamDirection[i].fluidPort_out,liquidCell
        [i + 1,hd.nPlateLength].fluidPort_in);
4  end for;
5  for j in 2:2:hd.n_passes - 1 loop
6    connect(liquidCell[j,1].fluidPort_out,
        liquidCellChangeOfStreamDirection[j].fluidPort_in);
7    connect(liquidCellChangeOfStreamDirection[j].fluidPort_out,liquidCell
        [j + 1,1].fluidPort_in);
8  end for;

```

---

Abbildung 8.4: Code: Verbindung *LiquidCell* mit *LiquidCellChangeOfStream*

Die Flüssigzellen sind an ihrer Oberseite mit dem  $\text{HeatPort}_{PE}$  der Elektronik verbunden, um so den entstandenen Wärmestrom abzutragen:



---

```

1 for i in 1:hd.n_passes loop
2   for j in 1:hd.nPlateLength loop
3     connect(heatPort_PE, liquidCell[i, j].heatPort_z);
4   end for;
5 end for;

```

---

Abbildung 8.5: Code: Verbindung LiquidCell mit Elektronik

Für die SolidCellEdge1 sind folgende Verbindungen hergestellt:

---

```

1 for i in 1:hd.nSolidEdge loop
2   for n in 1:hd.nPlateLength loop
3     for j in 1:hd.nPlateLength - 1 loop
4       connect(solidCellEdge1[i, j].heatPort_x_2, solidCellEdge1[i, j + 1].
5         heatPort_x_1) "connection between solid cells in a row";
6       connect(heatPort_ambientVertical_A_xz, solidCellEdge1[i, 1].
7         heatPort_x_1) "solid cells on left side on SolidEdge1 connected
8         to ambient";
9       connect(heatPort_ambientVertical_A_xz, solidCellEdge1[i, hd.
10         nPlateLength].heatPort_x_2) "solid cells on right side on
11         SolidEdge1 connected to ambient";
12       connect(heatPort_ambientVertical_A_yz, solidCellEdge1[1, n].
13         heatPort_y_2) "solid cells on upper side on SolidEdge1 connected
14         to ambient";
15       connect(heatPort_ambientHorizontal, solidCellEdge1[i, n].heatPort_z_1)
16         "all solid cells connected to ambient on bottom";
17     end for;
18   if hd.nSolidEdge == 1 then
19     connect(solidCellEdge1[1, n].heatPort_y_1, liquidCell[1, n].
20       heatPort_y_2) "down side of cell connected to liquid";
21   elseif hd.nSolidEdge == 2 then
22     connect(solidCellEdge1[1, n].heatPort_y_1, solidCellEdge1[2, n].
23       heatPort_y_2) "connection of the two solid cells";
24     connect(solidCellEdge1[2, n].heatPort_y_1, liquidCell[1, n].
25       heatPort_y_2) "connection of down side of Solid Edge1 to liquid";
26   else
27     for m in 1:hd.nSolidEdge - 1 loop
28       connect(solidCellEdge1[m, n].heatPort_y_1, solidCellEdge1[m + 1, n].
29         heatPort_y_2) "connection between the solid cells";
30       connect(solidCellEdge1[hd.nSolidEdge, n].heatPort_y_1, liquidCell[1, n]
31         ].heatPort_y_2) "connection of down side of Solid Edge1 to
32         liquid";
33     end for;
34   end if "hd.nSolidEdge >=3";
35 end for;

```

---

Abbildung 8.6: Code: Verbindung SolidCellEdge1

Dabei werden von Zeile 4 bis 8 im Code 8.6 zunächst die Zellen in einer Zeile nebeneinander verbunden und die Randzellen mit den **HeatPorts** der Umgebung verbunden. Zusätzlich könnte man den **HeatPort**<sub>SC<sub>Ez</sub>2</sub> der Randelemente mit der Konvektion der Umgebung oder der Elektronik verbinden.

In Zeile 10 bis 22 werden die Zellen in der Spalte und mit dem ersten Kühlkanal der **LiquidCell** an dem **HeatPort**<sub>LC<sub>y</sub>2</sub> miteinander verbunden. Dabei wird unterschieden ob die Diskretisierung von  $n_{SolidEdge}$  1, 2 oder größer ist. Ist die Diskretisierung 1, dann muss die eine **solidCellEdge1** Zelle mit ihrem **HeatPort**<sub>SC<sub>Ey</sub>1</sub> an dem **HeatPort**<sub>LC<sub>y</sub>2</sub> der Flüssigzelle verbunden werden. Ist  $n_{SolidEdge} = 2$  dann muss auch unter den **SolidEdge** Zellen eine Verbindung hergestellt werden, bevor die zweite Zeile der **SolidEdge** Zellen mit dem Kühlkanal verbunden wird. Bei  $n_{SolidEdge} \geq 3$  verläuft die Verbindung der Zellen untereinander in der Spalte durch die *for*- Schleife in Zeile 16.

Die Elemente der **solidCellEdge2** Matrix werden ähnlich miteinander verbunden, wie die der **SolidEdge1** Matrix. Hier sind die Elemente sozusagen nur spiegelverkehrt. Dadurch wird nicht die unterste Zeile der Matrix mit dem Kühlkanal verbunden, sondern die Verbindung erfolgt an der obersten Zeile mit dem **HeatPort**<sub>SC<sub>Ey</sub>2</sub> zur untersten Zeile der Matrix der Flüssigelemente am **HeatPort**<sub>LC<sub>y</sub>1</sub>. Ebenso ist auch die Verbindung zur Umgebung bei **SolidEdge2** an den Elementen der untersten Zeile.

Die Verbindung der Elemente des Zentrums in der Zeile und zur Konvektion der Umgebung erfolgt wie bei den **SolidCellEdge** Elementen. Lediglich die Verbindung zur Konvektion an der Seite  $A_{SCCyz}$  entfällt und die Zentrumszellen sind alle mit ihrem **HeatPort**<sub>SCCz2</sub> mit der Elektronik verbunden.

Die Verbindung zwischen den Zeilen und zu den Flüssigzellen ist bei den Zentrumszellen von der Diskretisierung  $n_{SolidCenter}$  und von der Anzahl der Kühlkanäle  $n_{passes}$  abhängig. Um zu unterscheiden in welchem Zentrum die Verbindung stattfindet, erfolgt die Deklaration abhängig von der Anzahl der Kühlkanäle. Dadurch ist zum Beispiel die erste Zentrumszeile jeweils  $n_{SolidCenter}(k-1)+1$  wobei das  $k$  in einer *for*- Schleife bis  $n_{passes}-1$  läuft (siehe Zeile 4 Code 8.7). Dadurch ergibt sich zum Beispiel für  $n_{passes} = 3$  und  $n_{SolidCenter} = 2$  beim ersten Durchlauf von  $k$  des Codes 8.7 (Zeile 1 und 7) die Zeile 7 zu `connect(solidCellCenter[1,n].heatPorty2, liquidCell[1,n].heatPorty2)`, beim zweiten und letzten Durchlauf, da  $k = n_{passes} - 1$  zu `connect(solidCellCenter[3,n].heatPorty2, liquidCell[2,n].heatPorty2)`. Dadurch ist ersichtlich, dass durch diese Deklaration der Zellen, direkt in das zweite Zentrum gesprungen wird. Die `solidCell[2,n]` wird dabei in Zeile 9 mit dem zweiten Durchlauf des Kühlkanals verbunden.

---

```

1  for k in 1:hd.n_passes - 1 loop
2    for n in 1:hd.nPlateLength loop
3      if hd.nSolidCenter == 1 then
4        connect(solidCellCenter[hd.nSolidCenter * (k - 1) + 1,n].
5          heatPort_y_2,liquidCell[k,n].heatPort_y_1);
6        connect(solidCellCenter[hd.nSolidCenter * (k - 1) + 1,n].
7          heatPort_y_1,liquidCell[k + 1,n].heatPort_y_2);
8      elseif hd.nSolidCenter == 2 then
9        connect(solidCellCenter[hd.nSolidCenter * (k - 1) + 1,n].
10          heatPort_y_2,liquidCell[k,n].heatPort_y_1);
11        connect(solidCellCenter[hd.nSolidCenter * (k - 1) + 1,n].
12          heatPort_y_1,solidCellCenter[hd.nSolidCenter * (k - 1) + 2,n].
13          heatPort_y_2);
14        connect(solidCellCenter[hd.nSolidCenter * (k - 1) + 2,n].
15          heatPort_y_1,liquidCell[k + 1,n].heatPort_y_2);
16      else
17        for m in 1:hd.nSolidCenter - 1 loop
18          connect(solidCellCenter[hd.nSolidCenter * (k - 1) + 1,n].
19            heatPort_y_2,liquidCell[k,n].heatPort_y_1);
20          connect(solidCellCenter[hd.nSolidCenter * (k - 1) + m,n].
21            heatPort_y_1,solidCellCenter[hd.nSolidCenter * (k - 1) + m + 1,n].
22            heatPort_y_2);
23          connect(solidCellCenter[hd.nSolidCenter * (k - 1) + hd.nSolidCenter
24            ,n].heatPort_y_1,liquidCell[k + 1,n].heatPort_y_2);
25        end for;
26      end if;
27    end for;
28  end for;

```

---

Abbildung 8.7: Code: Verbindung **SolidCellCenter**

Wie bei den Randmatrizen wird bei den Zentrums-elementen unterschieden ob die Diskretisierung 1, 2 oder größer ist, da die Zellen im Falle bei  $n_{SolidCenter} = 1$  (siehe Zeile 3 Code 8.7) am Port  $y2$  mit der  $k$ -ten Flüssigzelle am Port  $y1$  und mit dem Port  $y1$  mit der  $k + 1$ -ten Flüssigzelle verbunden sind. Bei  $n_{SolidCenter} = 2$  sind die **SolidCenter** Elemente zusätzlich unter sich verbunden und bei größeren Diskretisierungen läuft die Verbindung der Zentrumszellen in einer *for*-Schleife ab (siehe Zeile 11-16 Code 8.7).

## 9 Diskussion und Bewertung der Ergebnisse

In diesem Kapitel werden die Zellen und das Modell des diskretisierten Wärmeübertragers validiert. Das Modell soll hinsichtlich der Rechenzeit und der Funktionalität überprüft werden.

Da es keine Versuchsdaten gibt, können die genauen Werte nicht validiert werden. Es können lediglich die Tendenzen und das Verhalten des Modells überprüft werden.

### 9.1 Validierung der Flüssigzelle

In diesem Abschnitt soll die Flüssigzelle genauer überprüft werden.

Dabei werden mehrere Flüssigzellen in Reihe geschaltet, welche am Beginn mit einer `LiquidSource` und am Ende mit einer `LiquidSink` aus der `FluidDynamics` Bibliothek verbunden werden, welche das Kühlmedium `PropylenGlycolWater47` mit einer Temperatur von  $10^{\circ}\text{C}$  mit einem Massenfluss von  $0,3\text{kg/s}$  hineinströmen lassen. Die Flüssigzellen werden dabei alle mit einer vorgeschriebenen Temperatur am `HeatPortLCz` verbunden. Im allgemeinen gilt, dass sich ein `HeatPort` adiabat verhält, wenn er nicht verbunden ist. Somit bekommt oder gibt er keine Wärmeströme von und nach außen ab. Dies gilt in diesem Fall für die `HeatPortsLCy1,2`.

Als Material für das Rohr wird Kupfer mit  $c_p = 0$  und  $\lambda = \textit{konstant}$  verwendet. Bei konstanter Temperaturbelastung von  $100^{\circ}\text{C}$  für alle Zellen, sollte die Temperatur des Kühlmittels exponentiell ansteigen. Da hier lediglich ein sehr kurzer Kanal betrachtet wird, ergibt sich ein linearer Verlauf (siehe Abb.9.1). Fließt das Kühlmittel durch die erste Zelle, erwärmt es sich und tritt bereits wärmer als am Eintritt in die nächste Zelle in der die Temperatur weiter ansteigt. Der gleiche Effekt ist in Abb. 9.2 zu beobachten, wenn die Zellen mit einer linear ansteigenden Temperatur belastet werden. Das heißt die erste mit  $15^{\circ}\text{C}$ , die zweite mit  $25^{\circ}\text{C}$  usw. Dann erwärmt sich das Kühlmittel nicht mehr exponentiell, sondern quadratisch.

Da die Temperaturbelastung im Fall des linearen Anstiegs wesentlich höher ist, kann die Temperaturdifferenz der beiden Fälle nicht miteinander verglichen werden.

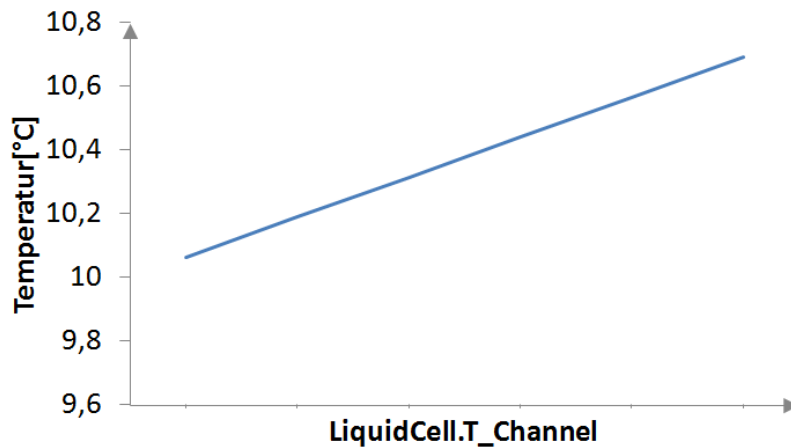


Abbildung 9.1: Validierung der Flüssigzelle: Exponentieller bzw. linearer Temperaturanstieg bei konstanter Temperaturbelastung

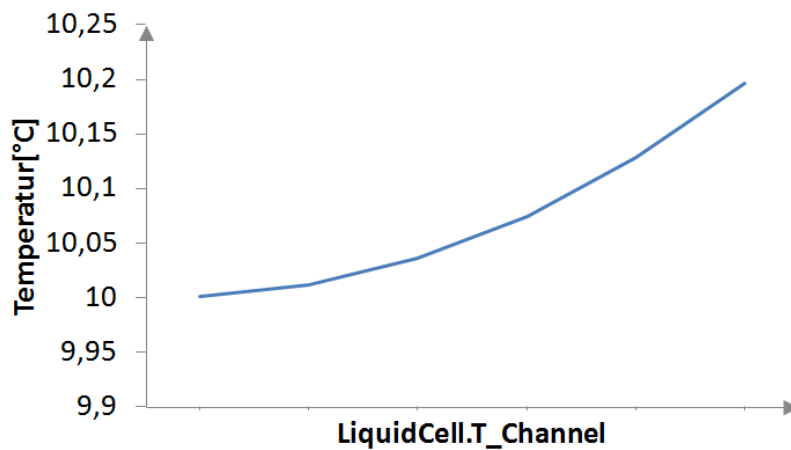


Abbildung 9.2: Validierung der Flüssigzelle: Quadratischer Temperaturanstieg bei linearer Temperaturbelastung

## 9.2 Untersuchung der verschiedenen Lösungsalgorithmen

Wie bereits im 2. Kapitel Software beschrieben, stehen in Dymola verschiedene numerische Lösungsverfahren zur Auswahl. Diese sollen hinsichtlich ihrer Rechenzeit und Genauigkeit überprüft werden. Getestet werden soll am dem Modell des diskretisierten Wärmeübertragers. Dabei werden, wie in Abb. 9.3, die `HeatPorts` der Umgebung jeweils mit einer Konvektion der Umgebung verbunden. Der `HeatPort` der Elektronik hat eine vorgegebene Temperatur von 80°C. `LiquidInlet` und `LiquidOutlet` werden jeweils mit einer `LiquidSource` und `LiquidSink` verbunden. Alle Größen wie Massenstrom, Eintrittstemperatur des Kühlmittels und Größe der „Cold Plate“ bleiben über die

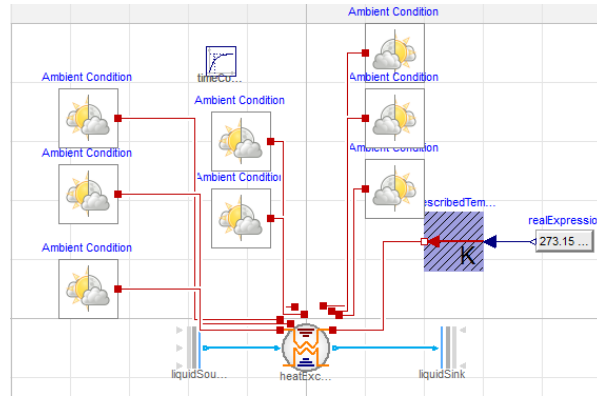


Abbildung 9.3: Testmodell

Simulationen konstant.

Es wird jeweils bei verschiedenen Diskretisierungen getestet,  $n_{passes} = 2$ ,  $n_{SolidEdge} = 1, 3, 5$  und  $n_{SolidCenter} = 1, 3, 5$ , wobei  $n_{SolidEdge}$  und  $n_{SolidCenter}$  jeweils immer gleich sind.

Vor der Durchführung der Simulation in Dymola kann im Simulationssetup das numerische Verfahren und die Start- und Stoppzeit ausgewählt werden, sowie die Anzahl der Zeitschritte, die zur Berechnung benötigt und ausgegeben werden sollen.

Interessant zu beobachten ist, dass alle Verfahren die gleichen Ergebnisse errechnen. Verglichen wurde dabei die Temperaturänderung des Fluides. Es kann folglich nicht gesagt werden, ob ein Verfahren genauer oder ungenauer rechnet.

Da als Medien jeweils Aluminium und Kupfer mit  $c_p = 0$  verwendet werden, wird sofort der stationäre Zustand erreicht. Es stellt sich aber dennoch die Frage, ob sich mit einer größeren Anzahl von Zeitschritten die Genauigkeit oder Rechenzeit ändert.

Allerdings bleiben auch hier die Ergebnisse gleich. Lediglich die Rechenzeit steigt, wodurch in diesem Modell eine größere Anzahl von Zeitschritten keinerlei Nutzen mit sich zieht. Wird jedoch nicht nur der stationäre Zustand betrachtet, kann man durch Erhöhung der Zeitschritte den Verlauf des Wärmeübergangs besser betrachten. Somit muss im folgenden nur noch auf die Rechenzeit der verschiedenen Löser geachtet werden, da sie alle die gleichen Ergebnisse liefern. Auffällig in Abb. 9.4 ist, dass der Standardlöser *Dassl* (Differential-Algebraic System Solver) von Dymola eine beinahe sechs mal längere Rechenzeit benötigt als das schnellste Verfahren *Radau IIa- order 5 stiff* (Radau Lobatto Formel Typ II A-stabiler Löser, ein 3 stufiges Runge-Kutta(RK) Verfahren).

Auch in der Literatur wird im Bereich der Fluidtechnik häufig das Verfahren *Radau IIa* empfohlen [27] [28]. Dass *Dassl* als Standardverfahren in Dymola gewählt wurde, liegt zum einen daran, dass Dymola dafür konzipiert wurde große Systeme zu modellieren. Da große Systeme häufig steif sind, wurde ein Verfahren gewählt, dass steife Modelle simulieren kann. Solche Verfahren können auch bei längeren Rechenzeiten nicht- steife Modelle simulieren. Anders herum ist dies nicht möglich. Zum anderen ist der Integrationsalgorithmus in einem Produktionscode implementiert, welcher zum Beispiel die Schrittweite regelt und die Einhaltung der Ordnung kontrolliert. Dieser Produktionscode nimmt ca. 95 % des Codes ein und der tatsächliche Algorithmus nur 5 %. Bei *Dassl* ist

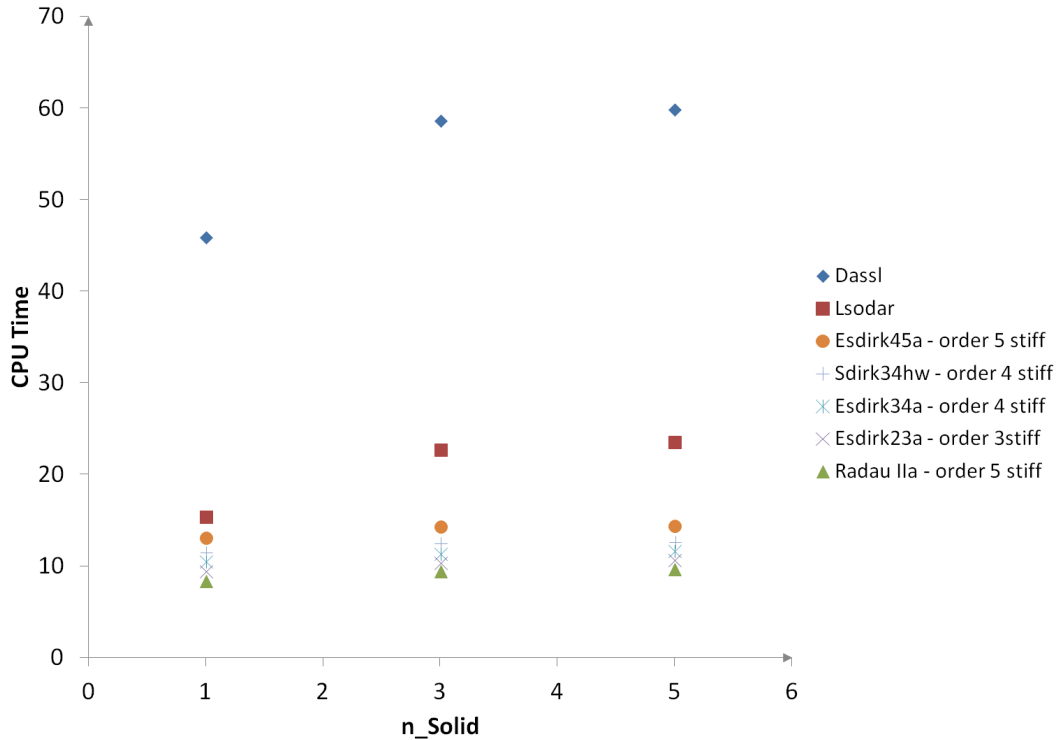


Abbildung 9.4: Rechenzeit von unterschiedlichen Lösungsverfahren bei verschiedenen Diskretisierungen

dieser Produktionscode besonders stabil und gut getestet, weshalb dieser Algorithmus als Standard gewählt wurde [29].

Interessant ist, dass die expliziten Verfahren *Euler*, *Runge-Kutta* (fix2, fix3, fix4), *Dopri45/8* (Dormand-Prince RK) und *Cerk23/45* (Continuous explicit RK) überhaupt nicht funktionieren und dabei die Simulation abgebrochen wird. Als Fehlermeldung wird dabei angezeigt, dass das nichtlineare System mit einem Newton-Löser nicht gelöst werden kann.

Die Verfahren *Sdirk34hw - order 4 stiff* (Singly diagonally implicit RK), *Esdirk45a - order 5 stiff* (Explicit singly diagonally implicit RK), *Esdirk34a - order 4 stiff* und *Esdirk23a - order 3 stiff* verhalten sich in Abb. 9.4 recht ähnlich. Diese unterscheiden sich allein durch ihre Ordnung. Dabei ist zu erkennen, dass eine höhere Ordnung eine größere Rechenzeit benötigt.

Das numerische Verfahren *Lsodar* (Livermore Solver for Ordinary Differential Equations with Automatic Method Switching and Root Finder) ist ein Verfahren, das automatisch zwischen steifen und nicht steifen Algorithmen wechseln kann. Dies erklärt auch, weshalb dieses Verfahren, nach *Dassl*, das langsamste ist, jedoch wesentlich besser mit den rein steifen Lösungsverfahren mithalten kann. Außerdem sind nur die Algorithmen *Dassl* und *Lsodar* Mehrschritt Verfahren, bei den anderen handelt es sich um Einzschritt Verfahren.

Dem Verhalten der integrierenden Löser nach zu urteilen, muss es sich bei dem Modell

„Cold Plate“ um ein steifes System handeln. Bei einem steifen System sind große und kleine Eigenwerte vorhanden. Es eignet sich ein Verfahren mit vielen Rechenschritten und mit kurzer Schrittweite. Dies sind vor allem implizite Verfahren [30]. Die Definition steifer Systeme nach [31]:

„Stiff equations are equations where certain implicit methods, in particular BDF, perform better, usually tremendously better, than explicit ones.“

Bei BDF Methoden handelt es sich um *Backward Differentiation Formulas*.

Demnach werden wegen der hohen Effizienz die folgenden Simulationen mit *Radau IIa* durchgeführt.

### 9.3 Validierung des adiabaten Wärmeübertragers

Um die Zellen und den Wärmeübertrager auf ihre Funktionalität hinsichtlich des Wärmeübergangs zu testen, wird das Testmodell aus Abb. 9.3 adiabatisch hinsichtlich der Umgebung getestet. Das heißt, dass das Modell nicht mit der Umgebung verbunden wird. Lediglich die Flüssigzellen und das Zentrum sollen mit der Elektronik verbunden sein. Somit muss die Summe der Wärmeströme der Zellen die mit der Elektronik verbunden sind die Gleiche sein, als das was die Elektronik abgibt. Außerdem muss der Wärmestrom um den sich die Flüssigkeit erwärmt, genauso groß sein, wie der Wärmestrom der von der Elektronik abgegeben wird, bzw. der von den Zellen aufgenommen wird.

$$\dot{Q} = \Delta h \dot{m} \quad (9.1)$$

Durch Einführung dieser Zeile in das Modell des diskretisierten Wärmeübertragers und durch händisches nachrechnen können diese Thesen überprüft werden. Wobei  $\Delta h =$

$\dot{m}$ [kg/s]	0,2	0,5	1	5
$\sum dh$ [J/kg]	33823,102	16974,983	12575,643	3633,1283
$\Delta h$ [J/kg]	33823,1	16975	12575,7	3633,1
$\dot{Q} = \Delta h \dot{m}$ [W]	6764,6204	8487,4915	12575,643	18165,6415
$\sum \dot{Q}$ [W]	6764,6225	8487,4922	12575,6399	18165,64
$\dot{Q}_{PE}$ [W]	6764,62	8487,49	12575,6	18165,6

Tabelle 9.1: Nachweis der gleichen Wärmeströme bei  $n_{Solid} = 3$  und  $n_{passes} = 3$

$(liquidOutlet.h_{outflow} - liquidInlet.h_{outflow})$  ist. Und  $\sum dh$  die Summe der Enthalpieänderung der einzelnen Zellen zwischen zwei **FluidPorts** ist.

Zu Tabelle 9.1 ist zu sagen, dass die gewählten Größen nicht an die Realität eines Wärmeübertragers angepasst sind und lediglich zeigen sollen, dass die Wärmeströme beim adiabaten System übereinstimmen.

An diesem Testmodell kann auch gut überprüft werden, dass die Festkörperzellen richtig arbeiten. Dabei werden die Verbindungen der **HeatPorts** betrachtet. Der Wärmestrom



der aus einem Port hinausgelangt, muss in den nächsten einfließen, und die Temperaturen an den verbundenen Ports müssen dieselben sein.

## 9.4 Überprüfung der Funktionalität der Diskretisierung

Zunächst sollen am oben genannten Testmodell (Abb. 9.3) verschiedene Diskretisierungen getestet werden. Die Anzahl der Durchläufe der Kühlkanäle variiert zwischen  $n_{passes} = 2 - 6$ . Die Diskretisierung der Festkörperzellen ist jeweils gleich und liegt zwischen von  $n_{Solid} = 1 - 12$ . Bei verschiedener Diskretisierung sollte die Temperaturänderung des Fluides, sowie der gesamte Wärmestrom gleich bleiben. In den beiden Abb.

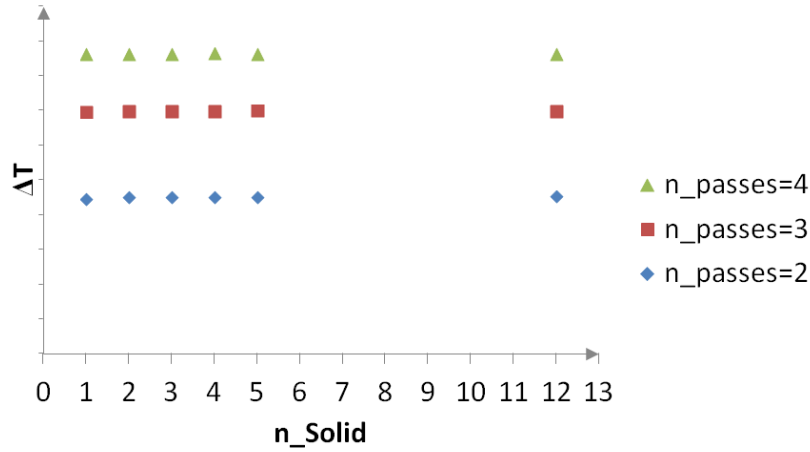


Abbildung 9.5: Überprüfung der Diskretisierung: Temperaturänderung  $\Delta T$  des Fluides über  $n_{Solid}$

9.5 und 9.6 ist gut zu erkennen, dass sowohl die aufgenommene Wärme als auch die Temperaturänderung  $\Delta T$  über die gleiche Anzahl der  $n_{passes}$  annähernd gleich bleiben. Bei Änderung der Anzahl der  $n_{passes}$  in Abb. 9.7 ändert sich auch die Anzahl der Zentren. Da die Flüssigzellen mit der Elektronik verbunden sind, besteht dadurch mehr Kontakt zur Elektronik, wodurch mehr Wärme von der Elektronik abgegeben und von der „Cold Plate“ aufgenommen wird. Dadurch wird die Änderung der Kühlmitteltemperatur  $\Delta T$  größer. Letztlich strebt aber auch dieser Effekt einem Grenzwert entgegen, da die Platte irgendwann nur noch aus Kühlkanälen bestehen wird, und die Größe des Zentrums insgesamt konstant bleiben wird. Bei genauer Betrachtung der Zelltemperaturen in Abb. 9.8a nimmt die gemittelte Zelltemperatur ab. Diese entspricht der Summe der Zelltemperaturen einer Spalte geteilt durch die Anzahl der Elemente. Zum Beispiel bei  $n_{Solid} = 3$  bildet sich die gemittelte Zelltemperatur der ersten Zelle durch  $(T[1, 1] + T[2, 1] + T[3, 1])/3$ . Ohne große Überlegungen würde man davon ausgehen, dass diese Temperatur unabhängig von der Diskretisierung gleich bleiben sollte. Abb. 9.8 zeigt allerdings Gegenteiliges,

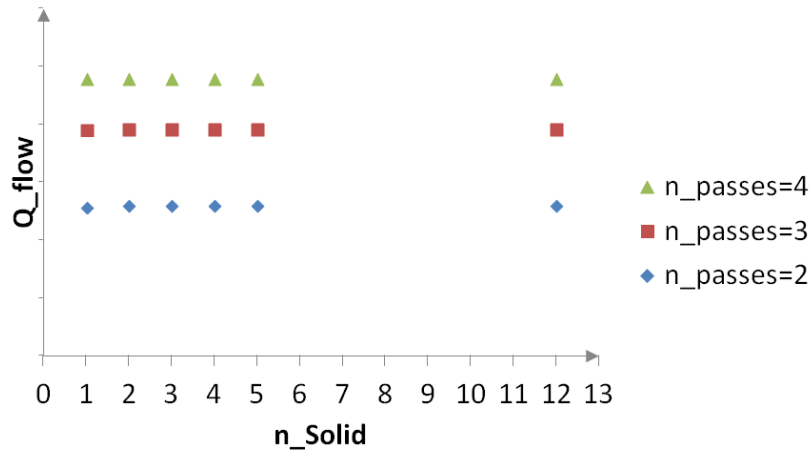
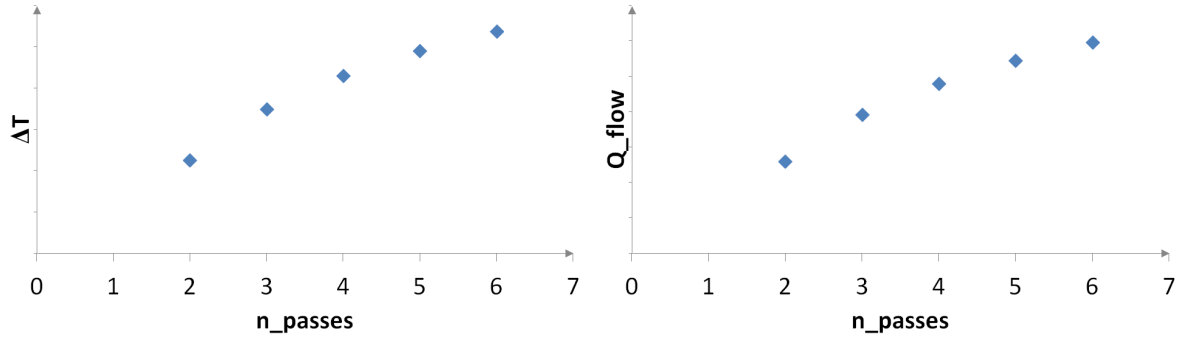
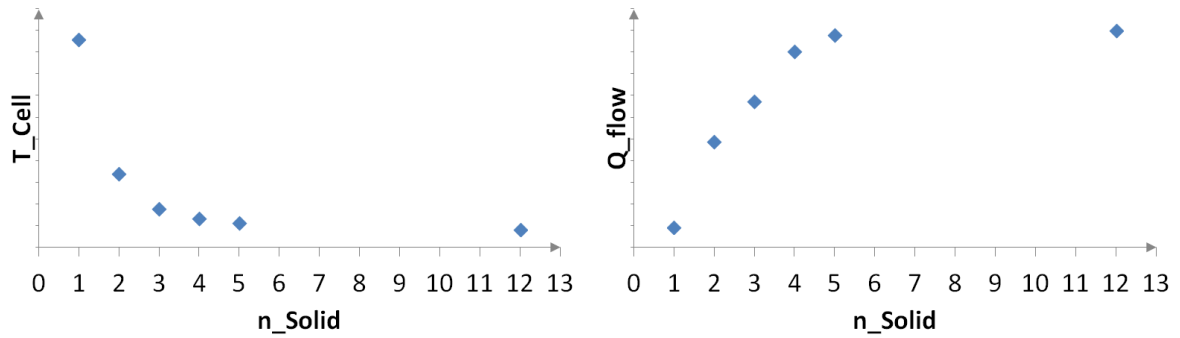

 Abbildung 9.6: Überprüfung der Diskretisierung: Wärmestrom  $\dot{Q}$  des Fluides über  $n_{\text{Solid}}$ 

 (a) Temperaturänderung  $\Delta T$  des Fluides über  $n_{\text{passes}}$  (b) Wärmestrom  $\dot{Q}$  des Fluides über  $n_{\text{passes}}$ 

 Abbildung 9.7: Überprüfung der Anzahl der Passes mit  $n_{\text{Solid}} = 5$ 


(a) Gemittelte Zelltemperatur

 (b) Wärmestrom  $\dot{Q}$  über  $n_{\text{Solid}}$ 

 Abbildung 9.8: Genauere Betrachtung der Diskretisierung bei  $n_{\text{passes}} = 2$

wobei die Temperaturschwankung zwischen dem höchsten und niedrigsten Wert bei nur 0,87°C liegen. Allerdings nähert sich die gemittelte Zelltemperatur einem Grenzwert an. Wie vermutet, gibt eine größere Diskretisierung realitätsnähere Wärmeübergänge wie-

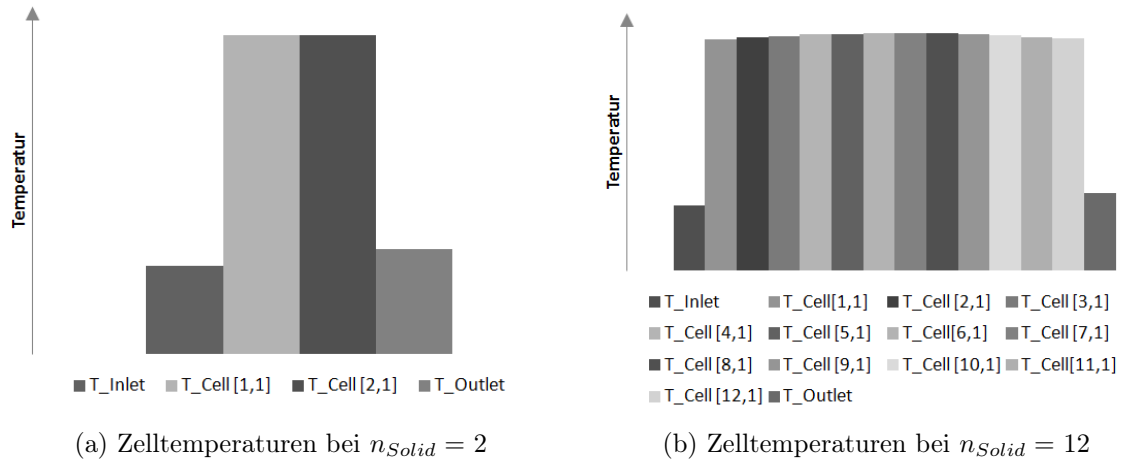
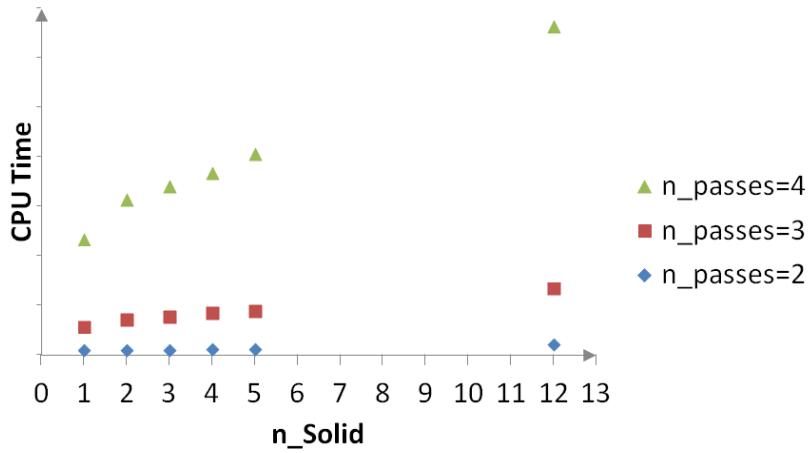
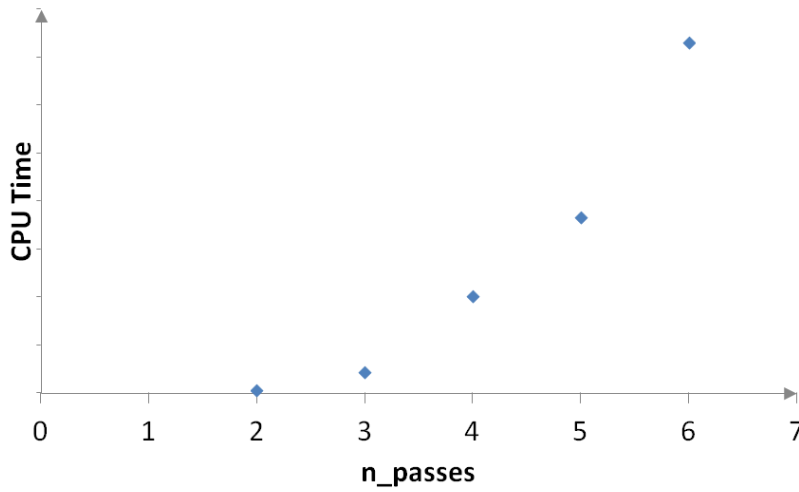


Abbildung 9.9: Zelltemperaturen zwischen zwei Kühlkanälen

der. Abbildung 9.9 zeigt die stufenförmige Verteilung der Zelltemperaturen. Der Verlauf in Abb. 9.9b entspricht eher der Realität eines Wärmeübergangs von einer umgekehrten Parabel als Abb. 9.9a. Man kann erkennen, dass die Temperatur der Zelle am **Outlet** größer ist, als derjenigen Zelle am **Inlet**. Schließlich ist auch die Temperatur des Ausgangs höher als die des Eingangs.

Da die gemittelten Zelltemperaturen über  $n_{Solid}$  abnehmen, steigt der Wärmestrom in Abb. 9.8b und somit auch das  $\Delta T$  des Fluides, da die Temperaturunterschiede zwischen der Elektronik und der Zelle größer werden (siehe Formel 5.3).

An dieser Stelle kann die Frage gestellt werden, was genau eine geeignete Diskretisierung ist. Dazu ist es nötig die Rechenzeit in Abb. 9.10 und Abb. 9.11 mit der Veränderung der Temperatur hinsichtlich anderer Diskretisierungen zu vergleichen. Da die Rechenzeit mit *Radau IIa* in Abb. 9.10 zwischen  $n_{Solid} = 5$  und 12 stark zu steigen beginnt und sich das  $\Delta T$  in Abb. 9.5 nicht mehr stark verändert, sollte hinsichtlich der Effizienz der Rechenzeit und Genauigkeit, die Diskretisierung in diesem Bereich liegen.

Abbildung 9.10: Rechenzeit CPU des Fluides über  $n_{Solid}$ Abbildung 9.11: Rechenzeit CPU des Fluides über  $n_{passes}$  bei  $n_{Solid} = 5$ 

## 9.5 Modell mit vorgeschriebenen Wärmestrom

In den vorherigen Validierungen ist die Wärmequelle der Leistungselektronik durch eine vorgeschriebene Temperatur modelliert worden. Hierbei verändern sich für verschiedene Diskretisierung und Anzahl der Durchläufe der Kühlkanäle der Wärmestrom  $\dot{Q}$  und somit die Temperaturänderung  $\Delta T$ . Bei einem vorgegebenen Wärmestrom bleibt  $\dot{Q}$  konstant. Der Wärmestrom teilt sich über alle Zellen die mit der Elektronik verbunden sind auf. Dadurch ist nicht wie in Abb. 9.8b, der Wärmestrom veränderlich sondern bei allen Diskretisierungen gleich. Folglich ergibt sich bei verschiedener Diskretisierung immer die gleiche Temperaturdifferenz  $\Delta T$  des Fluides (siehe Abb. 9.12).

Der Wärmestrom der Elektronik ist somit unabhängig von der Gleichung 5.3, wodurch er auch von der Temperatur der Festkörperzellen unabhängig ist. Dass in Abb. 9.13 dennoch die Temperaturänderung über  $n_{passes}$  steigt liegt wieder daran, dass mehr Flüssigzellen mit der Elektronik verbunden sind und somit die Wärme besser abtragen können. Interessant ist, dass die Rechenzeit bei vorgegebenen  $\dot{Q}$  wesentlich höher ist als

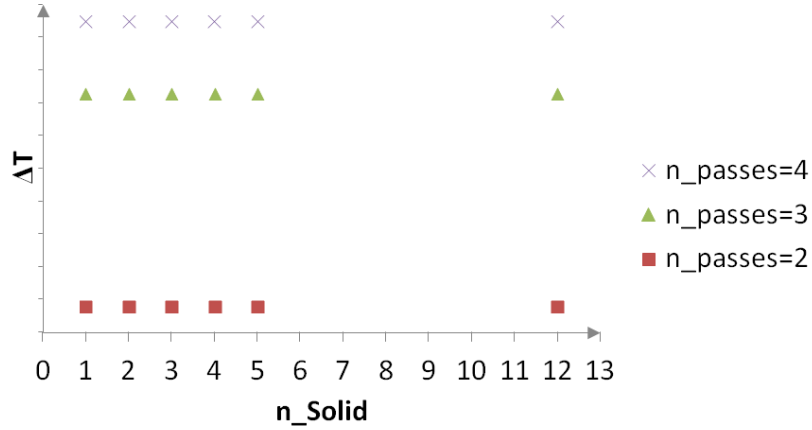


Abbildung 9.12: Temperaturänderung  $\Delta T$  bei vorgeschriebenen Wärmestrom  $\dot{Q}$  der Elektronik des Fluides über  $n_{Solid}$

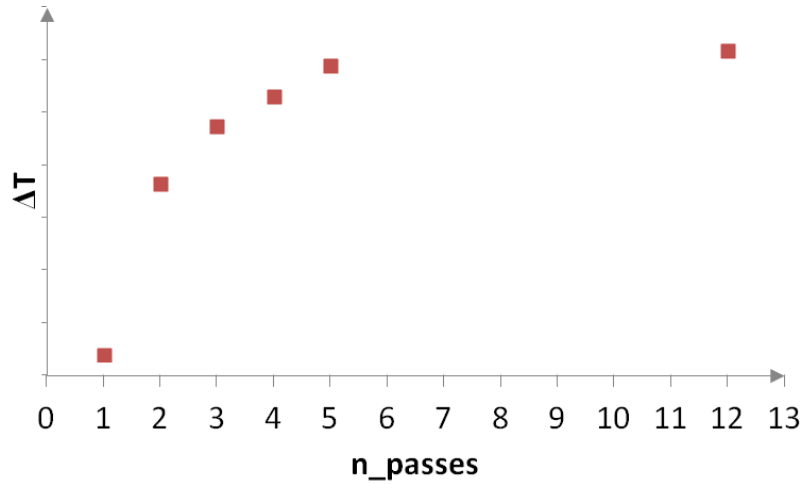


Abbildung 9.13: Temperaturänderung  $\Delta T$  bei vorgeschriebenen Wärmestrom  $\dot{Q}$  der Elektronik des Fluides über  $n_{passes}$  bei  $n_{Solid} = 1$

bei vorgegebenen  $T$ . Deshalb wurde in Abb. 9.13  $n_{Solid} = 1$  gewählt, und nicht wie im vorherigen Kapitel das Optimum mit  $n_{Solid} = 5$ . Bei Erhöhung des vorgeschriebenen  $\dot{Q}$  steigt die Rechenzeit, wobei ein Maximum erreicht wird, bei dem die Simulation nicht mehr durchgeführt werden kann. In diesem Fall lag der Wert bei 10000 W.

## 10 Visualisierung

Das Programm SimVis der Visualization Bibliothek wird dazu verwendet, Visualisierungen der „Cold Plate“ zu erstellen. Da Dymola zu jeder Simulation eine Matlab Datei mit den Simulationsergebnissen erstellt, können die Ergebnisse durch PostProcessing mittels der Funktion `readTrajectory` aufgerufen werden und der Funktion `setObjectArray` übergeben werden, welche eine Platte aus Kästchen aufbaut. Durch die Funktion `scalarToColor` wird ein skalarer Temperaturwert in eine Farbe umgewandelt, wodurch jedem Kästchen eine Farbe zugeordnet wird. Dazu wurde das Farbschema *jet* ausgewählt. Rot bedeutet heiß und blau kalt (s. Anhang Abb. .1). Als Beispiele sind in den Abb. 10.1 und 10.2 zwei Visualisierungen mit verschiedenen Diskretisierungen erstellt worden, bei denen die Temperaturveränderungen der Zellen über die Zeit zu sehen ist. Da sich die `LiquidCellChangeOfStream` außerhalb der Platte befindet ist diese nicht visualisiert. Die Größe der Kästchen und der Platte der Visualisierung entsprechen nicht der tatsächlichen Geometrie der „Cold Plate“. In den beiden Abb. 10.1 und 10.2 ist gut zu erkennen, dass sich bei größerer Diskretisierung eine realistischere Temperaturverteilung der Zentrumszellen ergibt. Die inneren Zellen der Diskretisierung sind dabei dunkler gefärbt als die äußeren, was bedeutet, dass sie höhere Temperaturen haben, da diese nicht in direktem Kontakt zu den Kühlkanälen stehen. Bei kleinerer Diskretisierung ergibt sich eher eine gleichmäßige Temperatur für alle Zellen (s. auch Abb. 9.9).

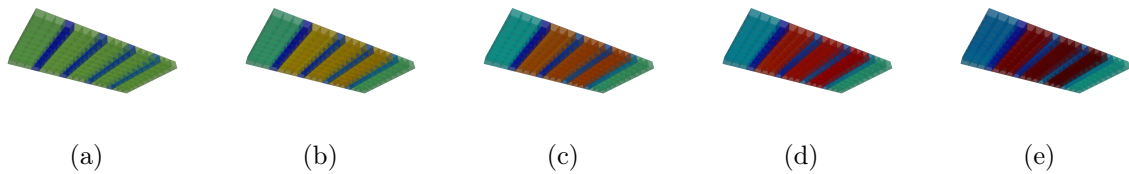


Abbildung 10.1: Visualisierung  $n_{Solid} = 3$  und  $n_{passes} = 4$

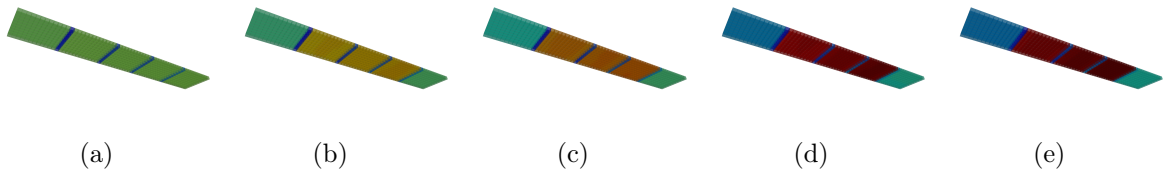


Abbildung 10.2: Visualisierung  $n_{Solid} = 12$  und  $n_{passes} = 4$

# 11 Zusammenfassung und Ausblick

Die Kühlung der Leistungselektronik in neuartigen Flugzeugkonzepten wie dem „More Electric Aircraft“ stellt neue Herausforderungen. Das in dieser Arbeit entwickelte Modell eines parametrisiert aufgebauten Wärmeübertragers in Form einer „Tubed Cold Plate“ soll der Vorauslegung von Wärmeübertragern für solche Flugzeugkonfigurationen dienen.

Durch die Einteilung des Wärmeübertragers in Flüssig- und Festkörperelemente und einer parametrisierten Verbindung der Elemente wird eine hohe Flexibilität erreicht. Es können Geometrien mit einer verschiedenen Anzahl von Verläufen des Kühlkanals bei unterschiedlicher Diskretisierung erstellt, und anschließend visualisiert werden, um die Temperaturveränderungen zu betrachten.

Dabei kann das Modell nicht nur hinsichtlich der Geometrie verändert werden. Durch die Erstellung einer Metallbibliothek stehen verschiedene Materialien für das Festkörperelement sowie für das Rohr des Flüssigelements bereit. Auch für das Kühlmedium kann auf unterschiedliche Medien aus einer DLR- internen Bibliothek zugegriffen werden.

Das Wärmeübertragermodell wird auch hinsichtlich der Umgebung modelliert. Dabei kann es entweder adiabat betrachtet werden, oder es stehen geeignete Korrelationen des Umgebungsmedium für die Konvektion bereit. Für den Druckverlust im Kühlkanal können Korrelationen gewählt werden.

Durch die Struktur des Modells des Wärmeübertragers, zum Beispiel der Zellen und der Geometrie in `partial models` und `records`, kann die Bibliothek des Wärmeübertragers einfach erweitert werden.

Das Modell wurde auf die Funktionalität hinsichtlich der Wärmeübergänge eingehend geprüft. Hinsichtlich der Rechenzeit ist überraschenderweise der Standardlöser *Dassl* von Dymola bis zu sechs mal langsamer als das integrierende Verfahren mit *Radau IIa- order 5 stiff*.

Die Frage einer geeigneten Diskretisierung ist zwischen Genauigkeit und Rechenzeit abzuwägen und wird dem Modellierer offen gelassen, obwohl bereits bei Diskretisierungen von  $n_{Solid} = 5$  gute Ergebnisse erzielt werden können.

Für weitere Arbeiten benötigt das in dieser Arbeit entwickelte Modell eine Validierung anhand von Versuchsdaten. Erst so können die implementierten Wärmeübergänge und Korrelationen für Konvektion und Druckverlust auf physikalischen Wahrheitsgehalt überprüft werden. Zusätzlich könnte auch Wärmestrahlung betrachtet werden.

Die Visualisierung kann noch erweitert werden, indem auch die Flüssigzelle mit der Änderung der Strömungsrichtung in die Visualisierung aufgenommen wird. Sowie, die Größe der Kästchen der tatsächlichen Größe des Wärmeübertragers angepasst wird.

Abschließend ist zu sagen, dass das Modell zur Vorauslegung von Wärmeübertragern geeignet ist, es aber erst noch einer Validierung anhand eines Testmodells bedarf.

# Literaturverzeichnis

- [1] Aavid Thermalololy. URL: <http://www.aavid.com/product-group/liquidcoldplates/extended>, Zugriff 09.2014.
- [2] Lytron Total Thermal Solutions. URL: <http://www.lytron.com/Cold-Plates>, Zugriff 09.2014.
- [3] Mattsson, S. E.; Elmqvist, H.; Otter, M. *Physical system moedling with Modelica*. Control Engineering Practise 6 (1998), 1997.
- [4] Mattsson, Sven Erik. *On modeling of heat exchangers in Modelica*. Proceedings of the 9th European Simulation Symposium, 1997.
- [5] Dietl, K.; Vasel, J.; Schmitz, G.; Casas, W.; Mehrkens, C. *Modeling of Cold Plates for Power Electronic Cooling*. Modelica Association, 2008.
- [6] Modelica. URL: <https://www.modelica.org/>, Zugriff 07.2014.
- [7] Hopfgarten, Siegbert. *Physikalisch objektorientierte Modellierung und Simulation mit Modelica/Dymola/OpenModelica*. TU Ilmenau.
- [8] Dassault Systems: Dymola. URL: <http://www.3ds.com/products-services/catia/capabilities/systems-engineering/modelica-systems-simulation/dymola>, Zugriff 07.2014.
- [9] Modelica Association. *Modelica - A Unified Object- Oriented Language for Physical Systems Modeling Tutorial*, 2000.
- [10] Bellmann, Tobias. *Interactive Simulations and advanced Visulization with Modelica*. Institute of Robotics and Mechatronics, German Aerospace Center (DLR).
- [11] Weigand, B. und von Wolfersdorf, J. *Wärmeübertragung Manuskript zur Vorlesung*. Universität Stuttgart ITLR, S. 1, 2007.
- [12] Weigand, B. und von Wolfersdorf, J. *Wärmeübertragung Manuskript zur Vorlesung*. Universität Stuttgart ITLR, S. 74,107, 2007.
- [13] Baehr, H. D. und Stephan, K. *Wärme- und Stoffübertragung*. Springer Verlag, S.455, 2010.



- [14] Weigand, B. und von Wolfersdorf, J. *Wärmeübertragung Manuskript zur Vorlesung*. Universität Stuttgart ITLR, S. 136, 2007.
- [15] Baehr, H. D. und Stephan, K. *Wärme- und Stoffübertragung*. Springer Verlag, S.381, 2010.
- [16] Weigand, B. und von Wolfersdorf, J. *Wärmeübertragung Manuskript zur Vorlesung*. Universität Stuttgart ITLR, S. 108, 2007.
- [17] Gnielinski, Volker. *G1 Durchströmte Rohre*. VDI Wärmeatlas, S. 787,788, 2013.
- [18] Knast, W. *L1 Druckverlust in durchströmten Rohren*. VDI Wärmeatlas, S. 1223, 2013.
- [19] Richter, Christoph C. *Proposal of New Object-Oriented Equation-Based Model Libraries for Thermodynamic Systems*. Dissertation: Fakultät für Maschinenbau der Technischen Universität Carolo- Wilhelmina zu Braunschweig, S. 61, 2008.
- [20] Idelchik, I.E. *Handbook of Hydraulic Resistance*. Begell House, 3. Auflage, S. 359, 1996.
- [21] Weigand, B. und von Wolfersdorf, J. *Wärmeübertragung Manuskript zur Vorlesung*. Universität Stuttgart ITLR, S. 10, 2007.
- [22] Casella, Francesco. *The Modelica.Fluid library*. Politecnico di Milano, S. 7.
- [23] Incropera, F. P. und De Witt, D. P. *Fundamentals of Heat and Mass Transfer*. John Wiley and Sons, 6. Auflage, S. 929ff, 2007.
- [24] Wagner, W.; Kretzschmar, H.-J. *D2 Stoffwerte von bedeutenden reinen Fluiden*. VDI Wärmeatlas, S. 197, 2013.
- [25] Pflugradt, Noah. *Dokumentation: Berechnungsprogramm feuchte Luft basierend auf Mollier-hx-Diagramm für feuchte Luft*. TU Chemnitz, S. 7, 2009.
- [26] Glück, Bernd. *Zustands- und Stoffwerte*. Verlag für Bauwesen GmbH, 2. Auflage, S. 88, 1991.
- [27] Beater, Peter. *Regelungstechnik und Simulationstechnik mit Scilab und Modelica*. Books on Demand GmbH, S. 205, 2010.
- [28] Berkeley Lab: Modelica Buildings Library. URL: <http://simulationresearch.lbl.gov/modelica/userGuide/bestPractice.html>, Zugriff 09.2014.
- [29] Cellier, F. E. und Kofman, E. *Continuous System Simulation*. Springer Verlag, S. 337, 2006.
- [30] Beater, Peter. *Regelungstechnik und Simulationstechnik mit Scilab und Modelica*. Books on Demand GmbH, S. 200, 2010.
- [31] Felgner, F.; Liu, L.; Frey, G. *Vergleich numerischer Löser zur Simulation steifer und hybrider Systeme*. Universität des Saarlandes, S. 4,6, 2010.

# Anhang

## .1 Metallbibliothek

Dieses Kapitel des Anhangs enthält die restlichen tabellarischen Stoffwerte nach [23] der Metallbibliothek, welche in der Arbeit noch nicht aufgelistet wurden.

### .1.1 Aluminium und Aluminiumlegierungen

$T [K]$	100	200	300	400	600
$\varrho [kg/m^3]$			2770		
$c_p [J/kgK]$	473	787		925	1042
$\lambda [W/mK]$	65	163		186	186

Tabelle .1: Stoffwerte von Aluminium 2024T6

$T [K]$	300	400	600
$\varrho [kg/m^3]$	2790		
$c_p [J/kgK]$	883		
$\lambda [W/mK]$	168	174	185

Tabelle .2: Stoffwerte von Aluminium 195 Guss

### .1.2 Kupfer und Kupferlegierungen

$T [K]$	200	300	400	600
$\varrho [kg/m^3]$		8800		
$c_p [J/kgK]$	785		460	545
$\lambda [W/mK]$	42		52	59

Tabelle .3: Stoffwerte von kommerzieller Bronze

$T [K]$	200	300	400	600
$\varrho [kg/m^3]$		8780		
$c_p [J/kgK]$		355		
$\lambda [W/mK]$	41		65	74

Tabelle .4: Stoffwerte von Phosphor Bronze

$T [K]$	100	200	300	400	600
$\varrho [kg/m^3]$			8530		
$c_p [J/kgK]$		360		395	425
$\lambda [W/mK]$	75	95		137	149

Tabelle .5: Stoffwerte von Messing

$T [K]$	100	200	300
$\varrho [kg/m^3]$			8920
$c_p [J/kgK]$	237	362	384
$\lambda [W/mK]$	17	19	23

Tabelle .6: Stoffwerte von Konstantan

### .1.3 Eisen und Eisenlegierungen

$T [K]$	100	200	300	400	600	800	1000	1200	1500
$\varrho [kg/m^3]$			7870						
$c_p [J/kgK]$	216	384		490	574	680	975	609	654
$\lambda [W/mK]$	134	94		69,5	54,7	43,3	32,8	38,3	32,1

Tabelle .7: Stoffwerte von reinem Eisen

$T [K]$	100	200	300	400	600	800	1000	1200	1500
$\varrho [kg/m^3]$			7870						
$c_p [J/kgK]$	215	384		490	574	680	975	609	654
$\lambda [W/mK]$	95,6	80,6		65,7	53,1	42,1	32,3	28,7	31,4

Tabelle .8: Stoffwerte von Armco

**.1.4 Silber**

$T [K]$	100	200	300	400	600	800	1000	1200
$\varrho [kg/m^3]$			10500					
$c_p [J/kgK]$	187	225		239	250	262	277	282
$\lambda [W/mK]$	444	430		425	412	396	379	361

Tabelle .9: Stoffwerte von Silber

**.1.5 Stahl und Stahllegierungen**

$T [K]$	300	400	600	800	1000
$\varrho [kg/m^3]$	7854				
$c_p [J/kgK]$		487	559	685	1169
$\lambda [W/mK]$		56,7	48	39,2	30

Tabelle .10: Stoffwerte von gewöhnlichen Stahl

$T [K]$	300	400	600	800	1000
$\varrho [kg/m^3]$	7832				
$c_p [J/kgK]$		487	559	685	1168
$\lambda [W/mK]$		58,7	48,8	39,2	31,3

Tabelle .11: Stoffwerte von AISI 1010

$T [K]$	300	400	600	800	1000
$\varrho [kg/m^3]$	7817				
$c_p [J/kgK]$		501	582	699	97
$\lambda [W/mK]$		49,8	44	37,4	29,3

Tabelle .12: Stoffwerte von Kohlenstoff Silizium

$T [K]$	300	400	600	800	1000
$\varrho [kg/m^3]$	8131				
$c_p [J/kgK]$		487	559	685	1090
$\lambda [W/mK]$		42,2	39,7	35	27,6

Tabelle .13: Stoffwerte von Kohlenstoff Mangan Silizium

## .1.6 Edeltahllegierungen

$T [K]$	300	400	600	800	1000
$\varrho [kg/m^3]$	8055				
$c_p [J/kgK]$		512	559	585	606
$\lambda [W/mK]$		17,3	20	22,8	25,4

Tabelle .14: Stoffwerte von AISI 302

$T [K]$	100	200	300	400	600	800	1000	1200	1500
$\varrho [kg/m^3]$			7900						
$c_p [J/kgK]$	272	402		515	557	582	611	640	682
$\lambda [W/mK]$	9,2	12,6		16,6	19,8	22,6	25,4	28	31,7

Tabelle .15: Stoffwerte von AISI 304

$T [K]$	300	400	600	800	1000
$\varrho [kg/m^3]$	8238				
$c_p [J/kgK]$		504	550	576	602
$\lambda [W/mK]$		15,2	18,3	21,3	24,2

Tabelle .16: Stoffwerte von AISI 316

$T [K]$	300	400	600	800	1000
$\varrho [kg/m^3]$	7978				
$c_p [J/kgK]$		513	559	585	6062
$\lambda [W/mK]$		15,8	18,9	21,9	24,7

Tabelle .17: Stoffwerte von AISI 347

## .1.7 Silizium

$T [K]$	100	200	300	400	600	800	1000	1200	1500
$\varrho [kg/m^3]$			2330						
$c_p [J/kgK]$	259	556		790	867	913	946	967	992
$\lambda [W/mK]$	884	264		98,9	61,9	42,2	31,2	25,7	22,7

Tabelle .18: Stoffwerte von Silizium

## .2 Farbschema Jet

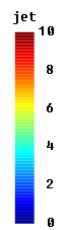


Abbildung .1: Farbschema Jet. 10 bedeutet heiß, 0 kalt.